

Amiga at
Fall COMDEX

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource Volume 3 Number 1
US \$3.50 Canada \$4.50

Count on the Amiga

BSpread
Current Sheet: SampleSheet.bsp
Use Project Menus or Select a Cell.

	October		November		
	Income	Expenses	Income	Expenses	Totals
Income	2000.00		2000.00		*****
Commissions	1500.00		1750.00		*****
Interest	250.00		255.00		*****
Mortgage Payment		900.00		900.00	*****
Car Payment		250.00		250.00	*****
American Express		210.00		150.00	*****
Visa		30.00		75.00	*****
MasterCard		25.00		36.00	*****
Phone Company		16.75		27.75	*****
Electric Company		22.50		35.25	*****
Gas Company		41.60		52.30	*****
Total for Month	*****				
YTD Total	*****				

Money Mentor

Create Expense Budget

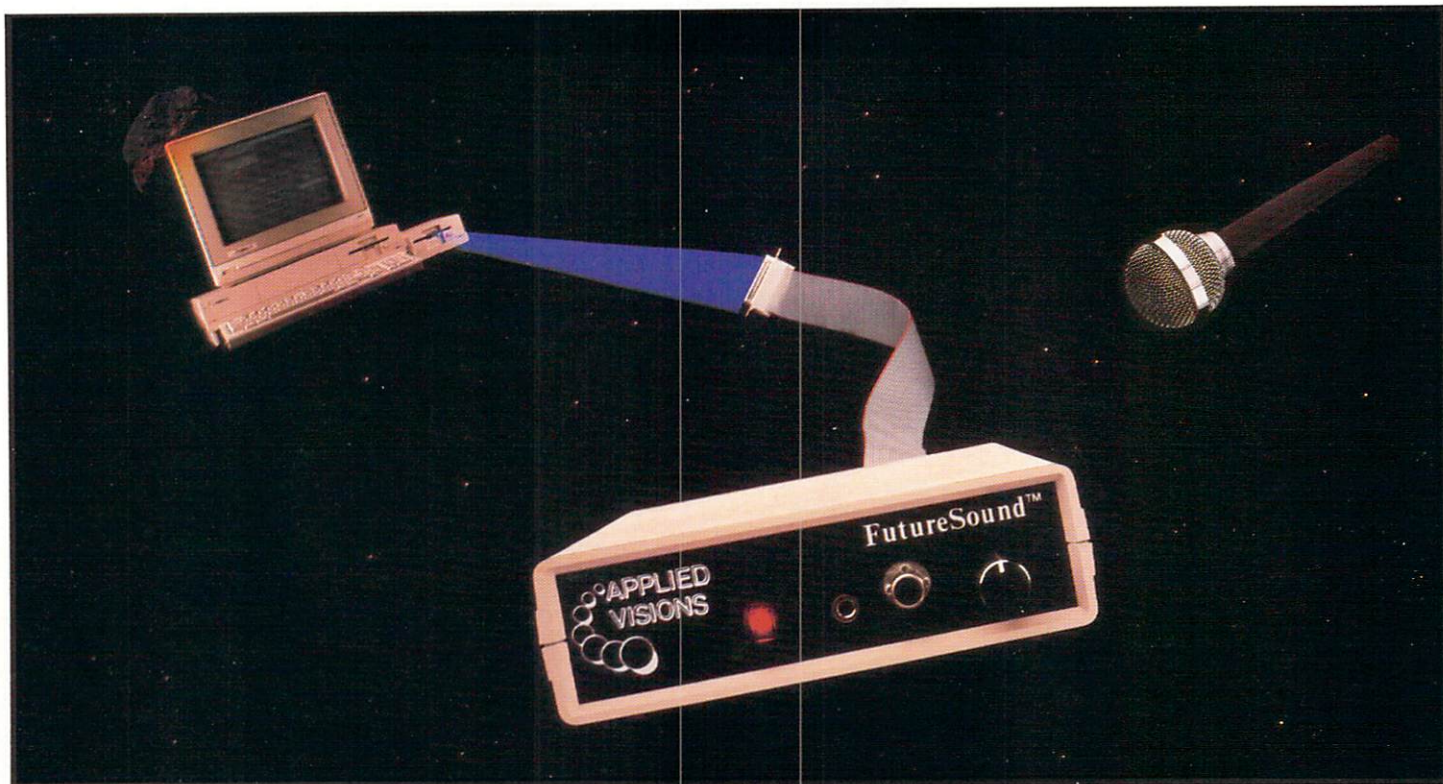
Expense	Jan	Feb	Mar	Apr	May	Year Total
Club Checking	2,000.00	2,000.00	2,000.00	2,000.00	2,000.00	24,000.00
Fica	46.50	46.50	46.50	46.50	46.50	550.00
GMC Car Payment	200.00	200.00	200.00	200.00	200.00	2,400.00
Groceries	90.00	90.00	90.00	90.00	90.00	1,079.10
Household	66.00	66.00	66.00	66.00	66.00	792.00
MasterCard	35.00	35.00	35.00	35.00	35.00	420.00
Medical	25.00	25.00	25.00	25.00	25.00	300.00
Mortgage	700.00	700.00	700.00	700.00	700.00	8,400.00
Other	100.00	100.00	100.00	100.00	100.00	1,200.00
Savings	500.00	500.00	500.00	500.00	500.00	6,000.00
Software Store	35.00	35.00	35.00	35.00	35.00	420.00
Taxes	110.00	110.00	110.00	110.00	110.00	1,320.00
VISA	36.00	36.00	36.00	36.00	36.00	432.00
Month Total	4,003.50	4,013.50	4,023.50	4,038.50	4,023.50	40,267.00

Value 700.00 F1 Edit F2 Copy F3 Repeat F4 Exit

with
BSpread:
The AmigaBASIC Spreadsheet
Investor's Advantage
Money Mentor
Haicalc



The Bottom Line on Spreadsheets



“Open the pod bay doors, HAL...”

Programmers cast their vote!

Right now, leading software developers are hard at work on the next generation of Amiga® products. To add the spectacular sound effects we've all come to expect from Amiga software, they are overwhelmingly choosing one sound recording package... FutureSound. As one developer put it, "FutureSound should be standard equipment for the Amiga."

FutureSound the clear winner...

Why has FutureSound become the clear choice for digital sound sampling on the Amiga? The reason is obvious: a hardware design that has left nothing out. FutureSound includes two input sources, each with its own amplifier, one for a microphone and one for direct recording; input volume control; high speed 8-bit parallel interface, complete with an additional printer port; extra filters that take care of everything from background hiss to interference from

the monitor; and of course, a microphone so that you can begin recording immediately.

What about software?

FutureSound transforms your Amiga into a powerful, multi-track recording studio. Of course, this innovative software package provides you with all the basic recording features you expect. But with FutureSound, this is just the beginning. A forty-page manual will guide you through such features as variable sampling rates, visual editing, mixing, special effects generation, and more. A major software publisher is soon to release a simulation with an engine roar that will rattle your teeth. This incredible reverberation effect was designed with FutureSound's software.



Question: What can a 300 pound space creature do with these sounds?

Answer: Anything he wants.

Since FutureSound is IFF compatible (actually three separate formats are supported) your sounds can be used by most Amiga sound applications. With FutureSound and Deluxe Video Construction Set from Electronic Arts, your video creations can use the voice of Mr. Spock, your mother-in-law, or a disturbed super computer.

Programming support is also provided. Whether you're a "C" programming wiz or a Sunday afternoon BASIC hacker, all the routines you need are on the non-copy protected diskette.

Your Amiga dealer should have FutureSound in stock. If not, just give us a call and for \$175 (VISA, MasterCard or COD) we'll send one right out to you. Ahead warp factor one!

Applied Visions, Inc., Suite 2200, One Kendall Square
Cambridge, MA 02139 (617) 494-5417

Amiga is a registered trademark of Commodore-Amiga, Inc.
Deluxe Video Construction Set is a trademark of Electronic Arts, Inc.



DIGI VIEW

DELIVERS ULTIMATE GRAPHICS POWER

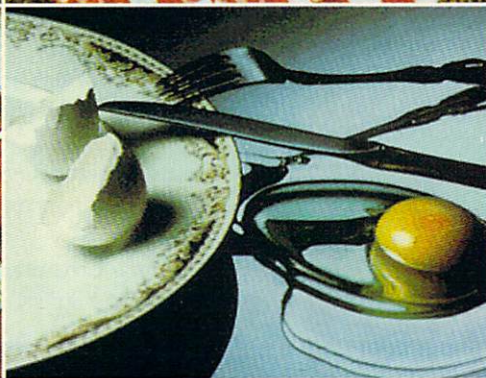
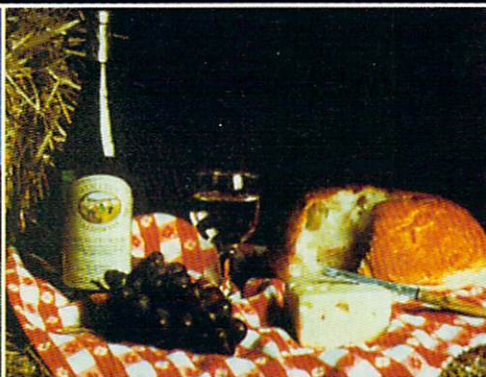
Bring the world into your Amiga with Digi-View, the 4096 color video digitizer. In seconds you can capture any photograph or object your video camera can see in full color and with clarity never before available on a home computer. Digi-View's advanced features include:

- Dithering routines give up to 100,000 apparent colors on screen
- NewTek's exclusive Enhanced Hold-and-Modify mode allows for exceptionally detailed images
- Digitize images in any number of colors from 2 to 4096
- Print, animate, transmit, store, or manipulate images with available IFF compatible programs
- Digitize in all Amiga resolution modes (320x200, 320x400, 640x200, 640x400)

"Digi-View sets new standards for graphics hardware"-InfoWorld

Digi-View is available now at your local Amiga dealer or call:
1-800-843-8934

ONLY \$199.95



NewTek
INCORPORATED

Amazing Columns

- AmigaNotes** by Richard Rae 51
Digital music generation on the Amiga.
- C Animation Part IV** by Michael Swinger 54
Just when you thought it was safe to go back in the C waters ...
- Forth!** by Jon Bryan 66
Sorting out CHIP and FAST memory on the Amiga.
- The Big Picture** by Warren Ring 69
Daring assembler language programming: CLI system calls and manipulating disk files.
- Bug Bytes** by John Steiner 77
Don't miss the latest upgrades and insidious bugs!
- Roomers** by The Bandito 90
AmigaDOS 1.3? 80386-based BridgeBoard for the A2000? and more!
- As I See It** by Eddie Churchill 91
Opinions, observations and the birth of a new software generation.
- 68000 Assembly Language Programming** by Chris Martin 93
"Create a multi-color screen without using Intuition routines."
- Modula-2 Programming** by Steve Faiwieszewski 99
A new contender bursts onto the Modula-2 scene!
- The Amicus Network** by John Foust 117
Public domain update, Commodore politics, and developer info.

Amazing Features

- Amicus Network Special Report: Fall COMDEX** by John Foust 18
Commodore at COMDEX and new products.
- The Ultimate Video Accessory: Part II** by Larry White 36
Spice up your videos with quality Amiga graphics.
- Life: Part II** by Gerald Hull 45
"A detailed look at efficient use of the Amiga blitter."
- FormatMaster: Professional Disk Formatting Engine** by Carl Mann 59
Put Batch language to work on the drudgery of disk formatting.
- BSpread** by Bryan Catley 79
A full-featured AmigaBASIC spreadsheet you can program!
- AmigaForum Transcript** ed. by Richard Rae 111
Zoom in on Commodore/Amiga's Dave Haynie.

Amazing Reviews

- Haicalc Review** by Chuck Raudonis 8
"A straightforward, easy to use, functional spreadsheet."
- VIP Professional Review** by Suzanne Mitchell 13
Easy stock portfolio management on the Amiga.
- Money Mentor Review** by Stephen Kemp 24
A personal finance system beyond your checkbook.
- Investor's Advantage Review** by Richard Knepper 28
plus "Poor Man's Guide to the Stock Market."

Amazing Departments

- From the Editor 4
Amazing Mail 6

- Public Domain Software Catalog 121
Index of Advertisers 126

35mm SLIDES FROM YOUR ARTWORK!



Professional 35mm Slides

- ◆ Now you can have reproduction and presentation quality slides of your work
- ◆ Distortion-free—fills in raster lines, crisp bright colors, converts all IFF files

Now Custom graphic art and illustration.

\$10 each for your 1st to 4th slides.
5 to 9 slides—\$8.50
Over 10 slides—\$8.00
Add \$2.00 for shipping.

New York residents add sales tax.

Call (212) 777-7609 FOR DETAILS

Ask for Ilene—or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276

Amazing COMPUTING™

Publisher:	Joyce Hicks
Circulation Manager:	Doris Gamble
Asst. to the Publisher:	Robert James Hicks
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble
Managing Editor:	Don Hicks
Submissions Editor:	Ernest P. Viveiros Jr.
Hardware Editor:	Ernest P. Viveiros Sr.
Music & Sound Editor:	Richard Rae
Art Director:	Keith M. Conforti
Advertising Manager:	John D. Fastino
Senior Copy Editor:	Michael T. Cabral
Copy Editor:	Karen P. Granger
Amicus & PDS Editor:	John Foust
Production Manager:	Mark Thibault
Production Assistant:	Rico A. Conforti
Production Assistant:	Keven Desmarais

Advertising Sales & Editorial
1-617-678-4200

Special thanks to:

Lynn Hathaway
Donna Peladeau
Traci Desmarais
Pilar Medeiros
Donna Thibault
Betsy Piper at Tech Plus.

Amazing Computing™ (ISSN 0886-9480)
is published by PIM Publications, Inc.,
P.O. Box 869, Fall River, MA 02722.
Subscriptions: in the U.S. 12 issues for
\$24.00; in Canada & Mexico, \$30.00;
Overseas: \$35.00. Printed in the U.S.A.
Copyright© 1987, 1988 by PIM Publications,
Inc. All rights reserved.

First Class or Air Mail rates available
upon request.

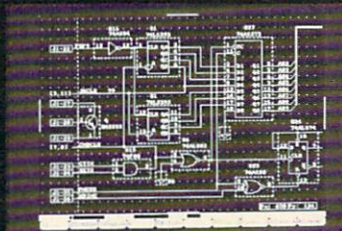
PIM Publications, Inc. maintains the right
to refuse any advertising.

PIM Publications, Inc. is not obligated to
return unsolicited materials. All materials
requesting return must be received with a
Self Addressed Stamped Mailer.

Prolific Inc.

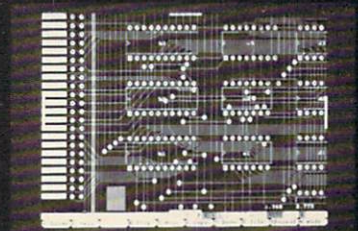
Available
Now!

PRO-NET



\$475.00

PRO-BOARD



\$475.00

For
AMIGA™
only

POWER+SIMPLICITY=PRODUCTIVITY

Feel our POWER...

PRO-NET:

- Variable template size
- A to E paper size
- Extensive Library included
- Auto device number with Zone control
- Gate swapping
- Single click change to negative logic
- Innovative weight assignment
- Auto page reference
- Dynamic error checking
- Creates BOM, Spare Part List, Net List, Error report, etc.
- Supports printers & plotters
- Supports Laser Printer
- Back annotation from PCB layout
- Move, Rubber... and lots more

See our SIMPLICITY...

Intelligent Function Keys make our programs extremely user friendly, provide maximum screen area, always display all relevant commands, avoids excessive cursor movement and screen flashing between menu & drawing, guides user through operation, minimizes training time.

PRO-BOARD:

- .025 inch grid
- .001 inch grid Library
- 12 mil trace, 13 mil space
- Produce 1, 2, 4 layer PCB
- Provides silk screen
- Auto coordinates assignment
- Auto produces power and grid layers
- Single line auto route
- Optional Net List Input for guided route, no need to look at schematic
- Prioritized route
- Dynamic error checking
- Supports printers, plotters and Gerber photo plotters
- Copy, Repeat... and lots more

**VISIT US AT L.A. AmiExpo
WESTIN BONAVENTURE HOTEL
404 S. Figueroa St., Los Angeles, CA
JAN. 16-18, 1988**

ORDER OR CALL FOR DETAILS
1808 W. Southgate Ave., Fullerton, CA 92633
Tel: (714) 447-8792 Telex: 5106016526 PROLIFIC CALIF
Western Union Easy Link Mail Box 62935949

TRY OUR DEMO DISKS FOR \$15 EACH.



Also from Prolific Inc.,

**4 full feature AMIGA™ Macro Cross Assemblers
for Z80, 6809, 8085 & 8051.**

See our SIMPLICITY

PRO-ASM

Feel our POWER

- Includes multi-pass Assembler, Linker, and Serial Down Load
- Generates relocatable Object Code Module
- Nested Macro
- Includes Files



\$85.00

Feel our POWER

- Conditional Assembling
- Rich set of directives
- Global and External Variables
- Data format includes Binary, Motorola Hex, Intel Hex, and Tek Hex

DEALER INQUIRY INVITED.

AMIGA™ trade mark of
Commodore Inc.

From The Editor:

Hello New Amiga Owners!

This issue of *Amazing Computing*™ should be seen by more new Amiga users than any previous issue. The reason is the increase in advertising produced through the Christmas season by Commodore Business Machines for the neatly packaged Amiga 500. The Amiga 500 is apparently Commodore's choice as a high-end "home" computer.

Although the advertisements appeared to concentrate heavily on the high graphics of the Amiga 500 (spelled "GAMES"), many consumers have been able to look past the obvious and view the Amiga as a tool for all types and styles of creativity. These new users are seeing the Amiga with many new wordprocessors, sound digitizing and music tools, calculating programs, databases, as well as graphics software now available.

The Amiga is a complete machine, offering many uses in a variety of directions. Whether you are using an Amiga 500 or the highly expandable Amiga 2000, you have a computer which has the opportunity to expand with you. But, nothing can happen until you know your machine.

Learn CLI

In the first issue of *Amazing Computing*™ (almost two years ago), we published, for the first time anywhere, a directory for the Command Line Interface. We followed with more instructions on moving through the subterranean layer of your Amiga. CLI is not simply an additional option for your Amiga; it is a serious disk operating system that lies beneath the casual and easy to use Workbench.

There are still a great many Public Domain Software disks and commercial programs that make use of the CLI. Even a simple file save, or directory search through a word-processor, requires a minimal knowledge of CLI.

Understand the Entire System

Do not stop at simply knowing how to master your Amiga's inner workings. You must also fully understand the workings of software and hardware peripherals. Mastering a new environment is what drives many users to spend late nights in front of a computer screen, and what inevitably causes some users to place their machines in a secure spot at the rear of the closet.

Becoming proficient at anything takes time and energy. Since the Amiga is a "personal computer," that commitment comes from you. Understanding will be a result of you learning a new approach to an old problem.

Define Your Problem

Concentrating on a small nuisance, which the computer can solve for you, gives you incentive to complete the task. Establish realistic goals and understand both the problem and your tools.

Some of the easiest, yet best, computer uses for a beginner include performing a list of activities, keeping a simple inventory or producing a small monthly mailing. By selecting a small problem and working through it, you will have shown a mastery of the Amiga, even on a simple level. From this success, you will want to attempt more difficult solutions. Be cautious and choose your problem well. Choosing increasingly difficult problems establishes a series of problems, solutions, and successes. Remember, "Nothing yields success like success."

Involve Your Family

The term Computer Widow (now also Widower) was coined several years ago with the first big push of computers into the consumer market. Men and women became so involved with their new toys, they fell victim to a narrow vision of the world. Their hours were spent between their com-

puters and all the things which got in the way of their computers. Spouses and small children beware! Naturally, this did not yield open acceptance of the computer by family members. Open resentment of the computer resulted, with a constant distrust of any new software or accessory, no matter how necessary.

The simple answer is to involve the family. With the Amiga, there are many levels of activity and ability. From simple music to advanced MIDI, wordprocessing to page layout, playing arcade games to designing arcade games, paint programs to CAD/CAM systems, simple spreadsheets to advanced relational databases, the Amiga offers the variety to keep any member of the family occupied with a love of his own. This will, however, require users to share, which can result in even more problems.

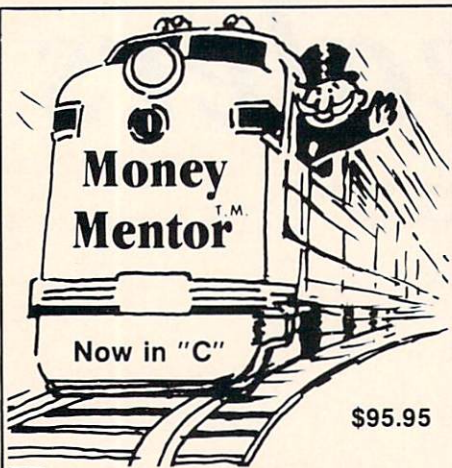
Involving other family members transforms the Amiga from an intruder to a family activity—a point of interest which can spark even better understanding between family members and create a reservoir of Amiga specialists in your home.

Amigas were developed as problem solvers. They are there to make living easier, to simplify tasks, and yield enjoyment. To use these tools, we must have a clear, uncluttered understanding of the tool, the problem, and our solution. Computers not only make people more productive, but they can make us better people.

Sincerely,

Don Hicks
Managing Editor

Amazing Computing™ is also available in most B. Dalton Booksellers stores, B. Dalton Software stores, Crown Books, and Software Etc. locations.



Money Mentor™ has a New Engine

Climb Aboard the new "C" version of Money Mentor™ for the ride of your life. Speed is your ticket to faster data input and dazzling graphics output. If your destination is better control of your personal finances, there's no faster way to get there than with Money Mentor™.

A unique system called "Smart Scrolls" handles a diversity of tedious data entry functions and can save 70% of the typing typically required for entry.

Money Mentor™ features:

- Net Worth Statement.
- 200 budget categories.
- 30 integrated accounts: checking, cash, saving and credit cards.
- Elaborate search routine allows editing of transactions according to your specific guidelines.
- Automatic check printing.
- Automatic Account Balancing.
- Colorful graphic reports illustrating actual versus budgeted amounts.
- Over 50 reports from which to choose.

**Let Money Mentor™ put your finances
on the right track... FAST!**



11844 Rancho Benardo Rd; Ste.#20
San Diego, CA 92128

To order,

call (619) 451-0151



Amazing Mail:

Dear Amazing Computing,

I was rather surprised to see that in your chart comparing word processors ["Not Three of a Kind" by Geoff Gamble, *Amazing Computing*, V2.11], you listed ProWrite as not being multitasking! Of course, it is multitasking. It throws up its own screen, and is quite well-behaved in its memory management. I have had V1.2 running simultaneously with WorkBench 1.2, Deluxe Paint1, and MaxiPlan, each of which has its own screens. I have also had it multitasking with PageSetter and MaxiPlan. I have 1.5 meg (via the Insider), and only when I run out of memory do I run into problems. Saving to ASDC's recoverable RAM disk vd0: while running these applications makes moving stuff between them pretty quick and easy. If that ain't multitasking, what is?

Hopefully in the future, software writers will incorporate testing for memory availability before trying to grab memory, so that lock-up or crashing won't be the norm for out-of-memory conditions on the Amiga. Currently I use ASLClockPlus, a PD program by Roy Laufer, which indicates both chip & expansion memory, as well as space still available on your disks, and date & time in a title bar-sized window. Although ProWrite and MaxiPlan have such a memory display feature, some of my software doesn't, so always knowing what's left comes in handy.

I'm a new subscriber, and very glad to see the quality coverage in your mag. Keep up the good work, and let's see some more Desktop Publishing articles.

Tom Roberts
Maine

Although at press time we were not able to contact Geoff Gamble, from the description of your system, it is possible to guess at the problem. Unfortunately, there is no real standard Amiga memory size. Each system can grow as large as the user needs (Amiga 500 memory expansions are now, or will soon be available).

This requires each user to pick his appropriate size. In the case of Mr. Gamble, we must assume he is using a "standard" 512K. This memory configuration surely constricts the multitasking potential of the program, resulting in the crashes reported.

*We will attempt to divulge the memory size of all authors' equipment, to give a more thorough view of a product's capability. The attempt to have every piece of extra hardware to make a particular piece of software shine in its best light is difficult. It is more fair to consider the user of a small system than the user of a large one. We would not want a user to purchase a piece of software, assuming his system would perform as stated in *Amazing Computing™*, and then be disappointed.*

Dear AC

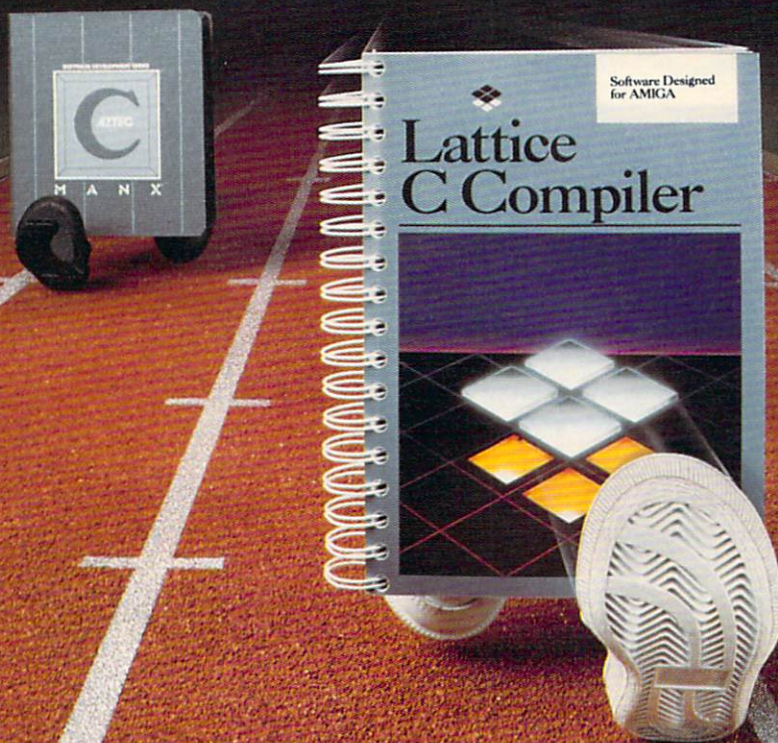
I have now read three issues of *Amazing Computing™*, Volumes 2.8 through 2.10, and have been thoroughly impressed by the content of the magazine. Several retail outlets now fly in *Amazing Computing™* and now we are able to get up to date information from the States.....

The only complaint that I originally had with *Amazing Computing™* was that it was printed mostly in Black and White, but you now seem to have rectified this. I don't know why you seem to think that some people would dislike this conversion, as it allows you to show the delights of true professional graphics capabilities on the Amiga. Although I suppose that it is similar to the controversy surrounding the colorization of B&W movies the industry gave to film buffs, if you don't like it, wear dark tinted sunglasses and you can have the old monochrome magazine back.

Keep up the good work.

Yours Faithfully
Danny Bielik
Sydney
New South Wales
Australia
•AC•

C who's winning the race. Lattice C for Amiga.



Lattice C has long been recognized as the best C compiler. And now our new version 4.0 for Amiga™ increases our lead past the competition even further.

Ready, set, go. The new Lattice AmigaDOS C Compiler gives you faster, more efficient code generation and support for 16 or 32-bit integers. There's direct, in-line interface to all Amiga ROM functions with parameters passed in registers. What's more, the assembler is fully compatible with Amiga assembler syntax.

More great strides. The linker, Blink, has been significantly enhanced and provides true overlay support and interactive recovery from undefined symbols. And you'll have a faster compile and link cycle with support for pre-linking.

There's no contest. Standard benchmark studies show Lattice to be the superior C language development environment. With stats like these, it's no wonder that Commodore-Amiga has selected Lattice C as the official Amiga development language.

Going the distance. You'll experience unsurpassed power and flexibility when you choose from several cost-effective development packages. There is even a full range of supporting products, including a symbolic debugger, resource editor, utilities and specialized libraries.

You'll discover that your software purchase is backed by an excellent warranty and skilled technical support staff. You'll appreciate having access to LBBS—one of the world's first 9600 baud, 24-hour bulletin board services. And you'll be able to conference with other Lattice users through the Byte Information Exchange (BIX) network.

Cross the finish line. Order your copy of the Lattice AmigaDOS C Compiler today. We'll supply the speed. You bring the running shoes.

	Lattice® Version 4.0	Manx® Version 3.40
Dhrystone	1294 Dhrystones/second	1010 Dhrystones/second
Float	22.20 Secs. (IEEE Format) 10.16 Secs. (FFP Format)	98.85 Secs. (IEEE Format) 17.60 Secs. (FFP Format)
Savage (IEEE)	47.67 Secs./000000318 Accuracy	119.6 Secs./000109 Accuracy

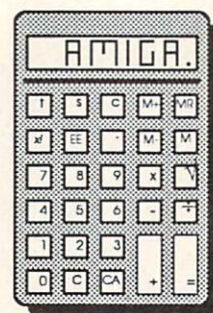

Lattice
 Subsidiary of SAS Institute Inc.

Lattice, Incorporated
 2500 S. Highland Avenue
 Lombard, IL 60148
 Phone: 800/533-3577
 In Illinois: 312/916-1600

haicalc

by Chuck Raudonis

SPREADSHEET



If you are one of the many who think they could benefit from a spreadsheet, but have been bewildered by the hundreds of commands used to run programs like Lotus 1-2-3 and Analyze!, here is just the program for you: Haicalc Spreadsheet from Haitex Resources. Haicalc does not claim to be Lotus 1-2-3 compatible or even a Lotus clone, as is the current spreadsheet rage. Haicalc is a straightforward, easy to use, functional spreadsheet.

The documentation is oriented towards the inexperienced user. It walks the user through all the steps and functions necessary to run the program. The documentation starts with a description of backing-up the system disk. Step one is "Turn on the Amiga." All the documentation is that thorough. It is written for all three Amigas, the 1000, 500, and 2000, and even anticipates the slight differences between machines.

The system disk is not copy-protected and the manual points out an implicit responsibility in this trust. The reasonable price of this program should be an incentive against piracy. If a program is useful, support the software developers. If they produce one useful program, you can be sure there are many good programs on the way, if the first one keeps them in business.

As I mentioned earlier, Haicalc does not intend to be a Lotus 1-2-3 clone. It is a fully-functional, "Amigaized" spreadsheet for the occasional spreadsheet user. He can balance the family budget or look at various financial options. Haicalc is also a well-behaved Amiga program. The program multitasks very well and an empty

spreadsheet. The windows are also linked. Although they operate independently, they all work on the same copy of the spreadsheet. If a change is made to a formula in one window, or if a piece of data is updated, the change is reflected in all windows. This option is very handy for doing "what if" analyses.

	JAN	FEB	MAR	APR	MAY	JUN
	\$111.00	\$123.00	\$951.00	\$54.00	\$666.00	\$221.00
	\$45.00	\$11.00	\$93.00	\$522.00	\$54.00	\$256.00
	\$45.36	\$54.00	\$85.00	\$278.00	\$36.00	\$5.00
	\$2135.05	\$5624.00	\$7778.00	\$64.00	\$652.00	\$4445.00
	\$123.00	\$284.00	\$54.00	\$325.00	\$88.00	\$23.00
	\$1.20	\$8.50	\$8.00	\$63.00	\$8.00	\$69.00
	\$0.25	\$2.50	\$7.00	\$98.00	\$55.00	\$548.00
	\$2221.00	\$555.00	\$89.00	\$5.00	\$84.00	\$3.00
Sun	\$4681.86	\$6654.00	\$9857.00	\$1489.00	\$1643.00	\$5570.00
Average	\$585.23	\$831.75	\$1132.13	\$176.12	\$285.38	\$696.25
Maximum	\$2221.00	\$5624.00	\$7778.00	\$522.00	\$666.00	\$4445.00

spreadsheet takes about 165K to run. As the spreadsheet is loaded, memory use goes up. Even on a 512K equipped Amiga, you can run two copies of the program to work on two spreadsheets at the same time.

In addition to multitasking, the program is also multi-windowing. It allows the user to open multiple windows in the same spreadsheet. This capability allows viewing of different portions of the same spreadsheet at the same time, and is very handy for comparison or analysis

This little bit of technical wizardry is accomplished by breaking the program into two main pieces. A "Cell" task runs in the background which manages the spreadsheet, performs all house-keeping, and maintains data for the spreadsheet. The other piece is the "Window" task which is responsible for interfacing with the user. The two tasks are interdependent; the Window task is the Cell task's only portal to the user. The Window task tells the Cell task how the user wants the data manipu-

lated. The Cell task then assumes responsibility for updating the data. This arrangement makes it possible to have multiple Window tasks all talking to the same background Cell task. Multiple windows can then view and update the same spreadsheet. This is a very clever design feature.

Haicalc also supports the full Amiga interface. It creates icons for its spreadsheets, but, due to the multiple task nature of the system, the program cannot be loaded by just clicking on the icon created for a spreadsheet.

(continued on page 10)

GOOD THINGS COME IN SMALLER PACKAGES.

The exciting NEW AVATEX 1200 E.



\$85
TOTAL DELIVERED
PRICE.

AVATEX
Authorized Sales &
Service

ACTUAL SIZE ONLY 5" X 6"

FREE! Communication Software & CompuServe Access Time with each MODEM.

We give you a
great price...

- + TOLL FREE ASSISTANCE
- + IMMEDIATE DELIVERY
- + NO CREDIT CARD SURCHARGE
- + NO SERVICE OR HANDLING FEES



FREE SHIPPING!

TOTAL HAYES COMPATIBILITY
8 LED STATUS INDICATORS
AUTO DIAL AND ANSWER
TONE OR PULSE DIALING

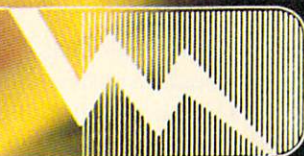
NEW 1200 E FEATURES

CCITT COMPATIBILITY
CALL PROGRESS DETECTION
INTERNAL SPEAKER
2 YEAR WARRANTY

Thousands of customers have purchased the AVATEX 1200 from Megatronics. New technologies now make it possible to offer an improved version of this popular Modem.

The 1200 E is amazing.
It weighs less and is smaller.
It has even more features.
It's still the same low price.

MEGATRONICS



MEGATRONICS, INC. P.O. BOX 3660, LOGAN, UT 84321

To Order

CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

800-232-6342

INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.

MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

The icons are used for file manipulations under the Workbench. The program does not replicate features already implemented in the Workbench. If you want to copy, rename, delete, or move a spreadsheet, these functions must be performed under the Workbench. This approach makes sense for the novice user; there is no need to learn a new way to delete a file. Just move it into the trashcan, and you are done with it.

Haicalc also supports full interlace mode and all Amiga fonts. An interesting aspect of this program is its ability to adjust to the chosen font's point size. While you can get 44 lines of text on the display using the interlace mode, the resulting flicker problem can be annoying. An easier solution is to use a 6-7 point font. This change results in a display of 27-30 lines of text, allowing you to see more of the spreadsheet while avoiding the ever-present interlace flicker.

In addition to the font support, the program also uses all available soft styles. Values and labels can be bold, underlined, italicized, etc. Unfortunately, this availability of fonts and softstyles does not extend to printed output. The output is a straight text dump, not a graphics dump. Accordingly, output is printed only in the font styles resident in the printer.

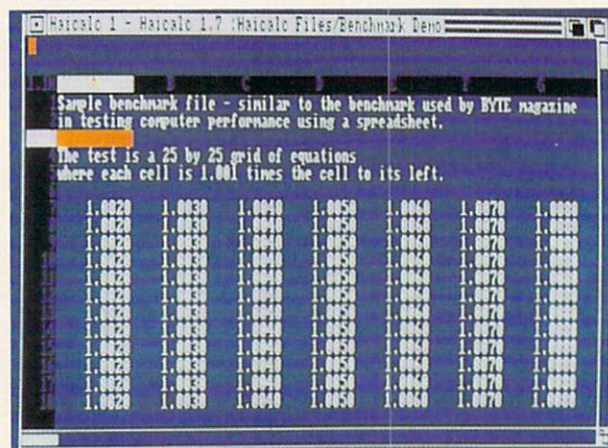
Haicalc also supports extended memory. Each window task has a "memory available" indicator. In the upper left corner of the spreadsheet, the system displays the amount of memory currently available for data use.

The mouse is a functional extension of the keyboard in running the program. The mouse moves the cursor around the screen, selects cells for use in calculations, and even selects multiple cells for cell range operations. The

latter operation is performed through an intuitive "click and drag" mouse operation.

All Haicalc functions are available through pull-down menus; there are no complicated control key combinations to remember. If something can be done, it appears on one of the menus. In addition, for the experienced user, all major functions are available as Amiga key combinations for ease of use.

Getting around the spreadsheet is very easy. The mouse and the cursor control keys allow you to navigate anywhere. The window task also has



proportional slider gadgets for moving quickly around the spreadsheet.

Haicalc has features found in most spreadsheet programs. A cell can contain a number, text (a "Label"), or a formula. Formulas are entered as they are entered in most spreadsheets. If you want to add the contents of cells A5 and B26, all you need to enter is +A5+B26. The results of the computation are then displayed in the current cell.

Haicalc is very weak in predefined functions. Most spreadsheets have a very rich set of functions available to the user. Haicalc has only has six pre-

defined functions: AVG, MAX, MIN, SUM, CELLS, and COUNT. The first four functions are self-explanatory; the last two require some explanation. COUNT returns the number of filled cells in a given cell range. CELLS returns the number of total cells in a given cell range (filled or unfilled). Most spreadsheets have functions for calculating present values and future values of funds, loan payment terms, and other financial functions. If you want to perform these functions in Haicalc, the full formulas must be entered as cell formulas. This hindrance is inconvenient at best. Usability is limited if you are not familiar with the formulas for these functions, but these functions are available in any good finance book.

Haicalc allows you to cut and paste any number of cells to duplicate them. The display format for values can be selected from a list of valid formats. This list includes zero, two, or four decimal positions in fixed decimal format, a dollars and cents and a straight dollar format, and a percentage format with zero or two decimal points (It is impossible to implement other configurations.). Text can be right, left, or center justified in columns. Column width can be set from one byte wide to more than a screen wide.

One very nice feature of Haicalc is its ability to adjust the sensitivity of the proportional gadgets. They can be adjusted to move over large or small areas of the worksheet. Another innovative addition is the ability to customize the function of the return key. The return key can be configured to act as a standard return key, or it can be assigned the function of an arrow key. If you are entering large amounts of data, you will appreciate this feature. The key can be set so the cursor follows the logical flow of the worksheet.

(continued on page 12)

AVAILABLE NOW! StarBoard2

If you've owned your Amiga® for a while now, you *know* you definitely need more than 512k of memory. You probably need *at least* double that amount...but you might need as much as an additional two megabytes. We want to urge you to use **StarBoard2** as the solution to your memory expansion problem –and to some of your other Amiga-expansion needs as well!

It's small, but it's BIG–

Since most of you want to expand your Amiga's memory without having to also expand your computer table, we designed **StarBoard2** and its two optional "daughterboards" to fit into a sleek, unobtrusive Amiga-styled case that snugly fastens to your computer with two precision-machined jackscrews.

The sculpted steel case of **StarBoard2** measures only 1.6" wide by 4.3" high by 10.2" long. You can access the inside of the case by removing just two small screws on the bottom and pulling it apart. We make **StarBoard2** easy to get into so that you or your dealer can expand it by installing up to one megabyte of RAM on the standard **StarBoard2** or up to two megabytes by adding in an Upper Deck.

This card has decks!

The basic **StarBoard2** starts out as a one megabyte memory space with 0k, 512k, or one megabyte installed. If you add in an optional **Upper Deck** (which plugs onto the Main Board inside the case) you bring **StarBoard2** up to its full two megabyte potential. You can buy your **StarBoard2** with the Upper Deck (populated or unpopulated) or buy the Upper Deck later as your need for memory grows.

And you can add other functions to **StarBoard2** by plugging in its second optional deck –the Multifunction Module!

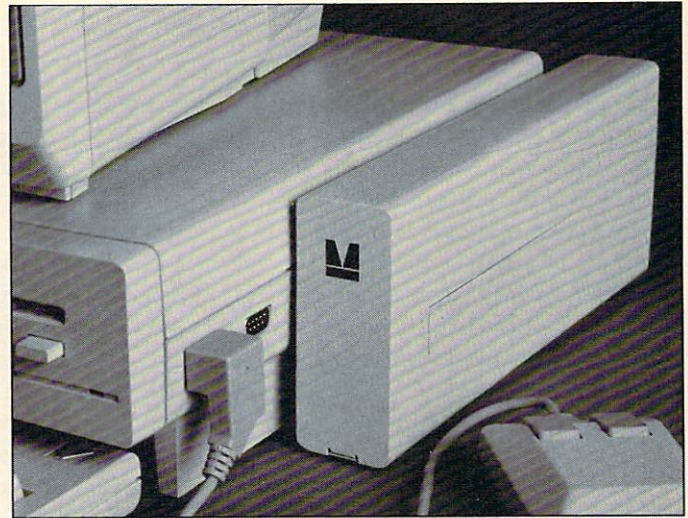
StarBoard2: functions five!

If we count Fast Memory as one function, the addition of the **Multifunction Module** brings the total up to five!

THE CLOCK FUNCTION:

Whenever you boot your Amiga you have to tell it what time it is! Add a Multifunction Module to your **StarBoard2** and you can hand that tedious task to the battery-backed,

**Auto-Configuring
Fast RAM
Zero Wait States
User Expandable
from 512k to
2 Megabytes
Bus Pass-Through
MultiFunction
Option: battery/
clock, FPU,
parity, Sticky-Disk**



real-time clock/calendar. A small piece of MicroBotics software in your WorkBench Startup-Sequence reads the clock and automatically sets the time and date in your Amiga. And the battery is included (we designed it to use an inexpensive, standard AAA battery which will last at least two years before needing replacement).

THE FLOATING POINT FUNCTION:

If any one aspect most characterizes the Amiga it's *fast* graphics! Most graphic routines make heavy use of the Amiga Floating Point Library. Replacing this library with the one we give you with your Multifunction Module and installing a separately purchased Motorola 68881 FPU chip in the socket provided by the Module will speed up these math operations from 5 to 40 times! And if you write your own software, you can directly address this chip for increased speed in integer arithmetic operations in addition to floating point math.

THE PARITY CHECKING FUNCTION:

If you install an additional ninth RAM chip for every eight in your **StarBoard2**, then you can enable *parity checking*. Parity checking will alert you (with a bus-error message) in the event of any data corruption in **StarBoard2**'s memory space. So what good is it to know that your data's messed up if the hardware can't fix it for you? It will warn you against saving that data to disk and possibly destroying your database or your massive spreadsheet. The more memory you have in your system the more likely it is, statistically, that random errors will occur. Parity checking gives you some protection from this threat to your data residing in Fast RAM. Note that the Amiga's "chip" RAM cannot be parity checked.

THE IMMORTAL MEMORY DISK FUNCTION (STICKY-DISK):

When you've got a lot of RAM, you can make nice big RAM-Disks and speed up your Amiga's operations a lot! But there's one bad thing about RAM-Disks: they go away when you re-boot your machine. Sticky-Disk solves that problem for you. It turns all of the memory space inside a single **StarBoard2**

into a Memory Disk that will survive a warm-reboot! When your Amiga attempts to grab a **StarBoard2** in Sticky-Disk mode, a hardware signal prevents the system from acquiring the **StarBoard2** as FastRAM (and thereby erasing your files) –instead it is re-recognized as a Memory Disk and its contents are preserved intact. If you want to work rapidly with large files of data that are being constantly updated (such as when developing software) you can appreciate the Sticky-Disk!

Fast RAM –no waiting!

StarBoard2 is a *totally* engineered product. It is a ZERO WAIT-STATE design, auto-configuring under AmigaDOS 1.2 as Fast RAM. Since AmigaDOS 1.1 doesn't support autoconfiguration, we also give you the software to configure memory in 1.1.

Any applications software which "looks" for Fast RAM will "find" **StarBoard2**. And you'll find that your applications run more efficiently due to **StarBoard2** on the bus.

A passing bus? Indeed!

What good is an Expansion Bus if it hits a dead end, as with some memory cards? Not much, we think –that's why we carefully and compatibly passed through the bus so you could attach other devices onto your Amiga (including another **StarBoard2**, of course!).

The sum of the parts...

A really nice feature of the **StarBoard2** system is that you can buy exactly what you need now without closing off your options for future expansion. You can even buy a 0k **StarBoard2** (with a one megabyte capacity) and populate it with your own RAM (commonly available 256k by 1 by 150ns memory chips). When you add **StarBoard2** to your Amiga you have a powerful hardware combination, superior to any single-user micro on the market. See your Authorized Amiga Dealer today and ask for **StarBoard2**

SUGGESTED RETAIL PRICING:	
StarBoard2, 0k (1 meg space):	\$349
StarBoard2, 0k (2 meg space):	\$395
StarBoard2, 512k (1 meg space):	\$495
StarBoard2, 1 meg (1 meg space)	\$595
StarBoard2, 2 megs installed:	\$879
StarBoard2, 2 megs & Multifunction:	\$959
Upper Deck, 0k (1 meg space):	\$ 99
Multifunction Module:	\$ 99
also available:	
Standard 256k memory card:	\$129
MAS-Drive20, 20 meg harddisk:	\$1495
MouseTime, mouseport clock:	\$ 50



MicroBotics, Inc.

811 Alpha Drive, Suite 335, Richardson, Texas 75081 / (214) 437-5330

AMIGA is a registered trademark of Commodore-Amiga

Haicalc can also print more than one range at a time (Most spreadsheets only print one range of data at a time.). Haitex realized that most spreadsheet applications require printing of multiple pieces of data simultaneously. The program lets you print several predefined ranges at the same time, saving you many repeated commands.

You are also not limited to the color choices made by the programmer. The entire color configuration is customizable, and your new configuration becomes the default.

In addition to the basic spreadsheet, Haicalc also has two graphing tasks: a pie chart and bar/line chart task. These two tasks also run as sub-tasks to the cell and window tasks. The main program must be running to run either graph task. The features and flexibility of these two charting options leave a lot to be desired. Only three sets of data can be graphed at the same time on the bar charts, and there

is no provision for customizing the graph. You cannot even print the graphs. The manual suggests a screen dump program (not supplied) for graph printing. While the graphing tasks might be suitable for some remedial analysis of data, they are not very flexible or useful.

The spreadsheet's most serious problem is precision. All calculations are performed in floating point. From the precision problems that are evident, it must be single precision floating point. The manual warns that numbers are only accurate to 8 positions. If a number greater than eight digits is entered, you can assume that only the first eight digits are accurate. If 1234567890 is entered into a cell, the system uses 1234567880.63 as the value for this cell. If the user is calculating household budgets, this should not be a problem, but if this package is being used to bid on commercial jobs, this limitation will certainly be a concern. In addition, if

a calculation requires many functions, a rounding error will be introduced with each calculation.

While Haicalc has its problems, it does have a market. If you are looking for a spreadsheet program for occasional home use, this program is ideal. It lacks the power of other Amiga spreadsheets, but, then again, it is only a fraction of the cost of other spreadsheets. This spreadsheet has the advantages of being simple to use and very easy to learn. If you have avoided spreadsheets because they were too complicated, take a look at Haicalc. It just might change your mind.

•AC•

haicalc™ Spreadsheet

List price \$59.95

Haitex Resources

208 Carrollton Park
Suite 1207
Carrollton, TX 75006
(214) 241-8030

ADD TO THE POWER OF YOUR PROGRAMS WHILE YOU SAVE TIME AND MONEY!

CBTREE does it all! Your best value in a B+tree source!

Save programming time and effort.

You can develop exciting file access programs quickly and easily because CBTREE provides a simple but powerful program interface to all B+tree operations. Every aspect of CBTREE is covered thoroughly in the 70 page Users Manual with complete examples. Sample programs are provided on disk.

Gain flexibility in designing your applications.

CBTREE lets you use multiple keys, variable key lengths, concatenated keys, and any data record size and record length. You can customize the B+tree parameters using utilities provided.

Your programs will be using the most efficient searching techniques.

CBTREE provides the fastest keyed file access performance, with multiple indexes in a single file and crash recovery utilities. CBTREE is a full function implementation of the industry standard B+tree access method and is proven in applications since 1984.

Access any record or group of records by:

- Get first
- Get previous
- Get less than
- Get greater than
- Get sequential block
- Get all partial matches
- Insert key and record
- Delete key and record
- Change record location
- Get last
- Get next
- Get less than or equal
- Get greater than or equal
- Get partial key match
- Get all keys and locations
- Insert key
- Delete key

Increase your implementation productivity.

CBTREE is over 6,000 lines of tightly written, commented C source code. The driver module is only 20K and links into your programs.

Port your applications to other machine environments.

The C source code that you receive can be compiled on all popular C compilers for the IBM PC and also under Unix, Xenix, and AmigaDos! No royalties on your applications that use CBTREE. CBTREE supports multi-user and network applications.

CBTREE IS TROUBLE-FREE, BUT IF YOU NEED HELP WE PROVIDE FREE PHONE SUPPORT.

ONE CALL GETS YOU THE ANSWER TO ANY QUESTION!

CBTREE compares favorably with other software selling at 2,3 and 4 times our price.

Sold on unconditional money-back guarantee.

YOU PAY ONLY \$99.00 - A MONEY-SAVING PRICE! NOW! Variable length records.

TO ORDER OR FOR ADDITIONAL INFORMATION

CALL (800) 346-8038 or (703) 847-1743

OR WRITE



Peacock Systems, Inc., 2108-C Gallows Road, Vienna, VA 22180

Give your stock portfolio the VIP treatment as fast as 1-2-3!

by Suzanne Mitchell

Move over, 1-2-3! VIP Professional is here to manage your stock portfolio via the Amiga. Every Amiga owner who has a stock portfolio should consider buying VIP Professional. This 1-2-3 mirror image does what it claims, with a fast calculation spreadsheet, database, and color graphics that won't quit.

Portfolio management can be time-consuming drudgery. This number-cruncher makes it fun—almost like playing with model trains. Even a modest portfolio deserves VIP Professional.

I prefer the spreadsheet approach to keep up with stocks and mutual funds on a weekly basis. Dedicated portfolio software gives me control over the format of my portfolio management; a spreadsheet, however, is flexible and gives me complete control over what is projected. I can set up the parameters and define the portfolio records in exactly the way I want them.

One drawback to the spreadsheet approach has been the lack of current security price retrieval from an on-line service such as CompuServe or Dow Jones. The Dow Jones Spreadsheet Link is now available for Lotus 1-2-3 (and any portfolio spreadsheet written with certain DJ Spreadsheet Link codes). On-line stock market information is interesting, but is more important to traders than to the average investor.

Anyone using the Fundamentalist approach to investing (i.e., evaluating and investing for the long haul), can easily track stocks using *The Wall Street Journal*, *Investors Daily*, *Value Line*, or a similar financial publication. This information is often available in libraries at no cost. You can keep track of the changing values of investments with weekly or monthly data updates and accurate tax records, without costly peripherals; spend your money on good stocks instead. With the loss of investment deductions under the new tax law, this is especially true.

The Mechanics of VIP

Using VIP requires some study, but it is time well spent. You may find the manual overwhelming at first. This manual has enough information to occupy you for the next 100 years, so be content to digest just a few basics initially. Practice working with the examples. If you are not familiar with VIP or Lotus 1-2-3 (version 1-A), read Nov./Dec. '86 *AmigaWorld* (p. 103) for an overview. Keep adding commands and simple formulas gradually to your repertoire. The ability to use VIP will be your reward!

Oh Where, Oh Where Did My Little RAM Go?

You need 512K of memory to run VIP. When you start the program, the screen displays 60K available memory if all the demo files have been deleted (56-58K with the demo files intact). This is my observation, not VIP

Technologies'. As I started exploring the more subtle mechanics of VIP, namely entering through CLI to gain extra memory, nothing happened. (A quick call to the company revealed there has been an update since my purchase last spring. The new update contains spiced-up demos and a new Wave graph using transcendental functions, an XY graph logo, and a graph for engineers.) After deleting all the demo files from the update copy, and booting through CLI, the available memory is 70K (worksheet bytes: 70632). Entering through the VIP icon produces 58K to 60K.

To gain these extra bytes, follow one of these procedures. With two disk drives, load Workbench, and at the AmigaDOS copyright, press (CTRL) and (D) simultaneously. At the 1> prompt, shrink the Workbench screen and type: `cd df1: [RETURN]`. At the 1> prompt, type: `Professional [RETURN]`. With one disk drive, change the directory to Professional; the name of the disk. Type `Professional`, press return, and the program will load. At the prompt, replace the Professional disk with Workbench. To get the maximum memory and bytes per spreadsheet, try resetting from Workbench to clear RAM.

How your spreadsheet was constructed affects how much memory is left to run the program. As you build a portfolio spreadsheet, keep the memory requirements in mind. Here are a few hints for conserving memory. Blank spaces take memory. Use

(continued)

VIP's "%" command instead of formulas using "*100." Eliminate everything not necessary for clarity. Get in the habit of checking STATUS along with FILE SAVE.

Rebooting (/FS, please!) after a "memory full" message often restores the RAM which you used up with "erase" and "delete" commands. The manual includes a procedure for clearing memory from the worksheet, but I find it easier to /FS, QUIT and reboot when worksheet bytes are low.

This list shows, in increasing order, how much RAM is required to display certain spreadsheet functions:

1. Simple numbers (integers) require the least memory.
2. Character Labels (ABCDEFGH, etc.).
3. Small Formulas (+D12..C12).
4. Large labels (ABCDEFGH, etc. from 50 characters up) require the most memory.

With some practice, you can bob-tail formulas to conserve memory. Here is an interesting example of how bytes can be saved by thinking out formula entries:

STOCKS	COST	% OF TOTAC	+B4/@SUM(B4..B7)
PE	1844.52	26%	C5 +B5/@SUM(B4..B7)
TED	2251.75	31%	C6 +B6/@SUM(B4..B7)
STRA	1542.00	22%	C7 +B7/@SUM(B4..B7)
TOFU	1524.00	21%	B9 @SUM(B4..B7)

7162.27

You can save over 300 bytes of memory by making those calculations with these formulas:

```
B4 +B4/B9
B5 +B5/B9
B6 +B6/B9
B7 +B7/B9
B9 @SUM(B4..B7)
```

May I Have a Transfer, Please?

Problem: You want to transfer an investment file on VIP copy B [df1] to VIP copy A [df0]. Answer: With copy A in df0, bring up a blank worksheet. Change the directory (/FD) to df1 [df1:]. Retrieve (/FR) the file you wish to copy from df1. The file in df1 will overlay the blank worksheet in df0. Change the directory (/FD) back to df0. From the df0 menu, execute FILE SAVE (/FS). The file from B [df1] is now copied and saved to disk A in df0.

As an investor, you are already using certain performance measurements to track stocks. There may be some things you would like to add. You'll ultimately save time and frustration if you give your AMIGA a rest, sit down at a desk, and begin the portfolio spreadsheet on paper. First, determine the number of stocks you want to monitor on one worksheet. A flow-chart type outline is helpful for headings, the order formulas must be entered, what data is necessary to include for calculations, and building the formulas.

Do not to use forward references. (A calculation which refers to a formula or data which has not been entered is a circular or forward reference.) VIP will alert you to this error by displaying an icon in the lower right screen or ERR in columns or rows.

The order of typical column headings would include Ticker symbol, date acquired, number of shares, purchase price, current price, and unrealized dollar gain or loss. These headings can be further refined to include percentage gain or loss, yield on cost, and yield on current value. I have deleted the Long/Short Term Capital gain column from my spreadsheets because it is redundant. (The 1987 tax law changes make it an unimportant entry now, anyway.) This frees up space and memory for other information such as what percentage a certain stock occupies in the portfolio or

annual dividend income. Daily, weekly, or monthly updates are automatic, once your template is set up. Enter the current stock price, and VIP takes over and updates your portfolio.

Here is a tip for entering fractions: Enter stock prices (17 5/8) as 17+5/8. This will display and print 17.625.

Besides printing out the spreadsheet, a hard copy of all cell entries is made from: "Print, Other, Cell-Formulas." Consult your printer manual for codes and enter the Print Menu. Under "OPTIONS, SET-UP," enter the code for your printer. Compressed mode for Epson LX80 is \015. The following set-up works on Epson MX:

```
\015 Print in compressed mode
\027\048 Print at 8 lines per inch
\027\050 Print at 6 lines per inch
\027\052 Print worksheet in italics
\027\069 Print emphasized mode
\027\071 Double strike
```

Did Auntie Em leave you a few shares of Dividend Reinvestment Plan (DRP) stock in her will? Or do you belong to an investment club which allows you to purchase of one share of a DRP stock? DRP stocks have been screened for future value and will be held for the long haul. Unless you are heir to a fortune (or stop buying software), small sums along with dividends will be invested regularly throughout the years until you decide to sell. This creates many stock purchase entries; you need to keep an accurate running average price for the stock. This average price (dollar cost averaging) will give you the base for gain or loss when the stock is finally sold. The IRS will be very impressed with your records in the event you are audited!

(continued on page 16)

SOURCE LEVEL DEBUGGER

Announcing the Manx Aztec C Source Level Debugger for the Amiga!

NOW THE MOST ADVANCED COMPUTER IN THE WORLD HAS THE MOST
ADVANCED SOURCE LEVEL DEBUGGER IN THE WORLD:

Manx Aztec SDB.



Save Time and Effort

If you're a pro at working with low level Amiga debuggers, you'll recognize what a source level debugger can mean—time savings. Time you'd rather spend creating than debugging. And if you're a beginner, SDB will make you a pro in no time.

Outstanding Features

That's why our new windowed SDB is so spectacular—because it's full of exciting features that make debugging a breeze. Of course, SDB has all of the features you expect from a debugger—like line-by-line tracing. Conditional breakpoints on lines, functions, or variables. Examination, modification, and display of global, local, and static variables, structures or expressions by name.

But SDB is also full of unexpected, incredibly sophisticated features. There's reusable command macros and procedures. Back tracing. Active frame context switching—just to name a few. Wait till you see SDB in action—it will blow you away!

Our Commitment to You

Manx Software Systems is the leading edge in C development systems. That means continual updates that bring the best to you.

Experience version 3.6 of Aztec C68k/Am today. And enjoy the most advanced debugger available ... on any computer.

- View your C Source
- Enter commands
- View your command output ...

... all at the same time!

Standard Features of Aztec C68k/Am 3.6:

- optimized C with selectable 68020 and 68881 support
- 680x0 Macro Assembler with 68881 support
- linker/librarian with overlays, scatter load, and segmentation
- symbolic debugger
- supports Amiga object format
- UNIX, AMIGA, and general purpose run time routines
- runs under CLI and supports all Workbench functions
- creates CLI and Workbench applications
- Supports both 1.1 and 1.2 Amiga DOS
- 600 pages of documentation and great example programs

Extended Features of Aztec C68k/Am 3.6

- UNIX utilities make, diff, grep, obj, ord, and vi
- special math support libraries for 68881 and Manx IEEE emulation

Portability: Aztec C is also available for the Macintosh, Apple II, MS-DOS/PC-DOS, CP/M-86, TRS-80, ROM, and others. Aztec C68k Third Party Software: A large array of support software is available for Aztec C68k. Call or write for information. The following is a list of the most requested products: Power Windows • Amiga View • Key to C • Amiga Lint • Metascope. Immediate Delivery Available for Most Destinations. Aztec C is available on a thirty day money-back guarantee. C.O.D., VISA, American Express, MasterCard, wire (domestic and international), and terms are available.

AZTEC C VERSIONS TO SUIT YOUR NEEDS:

Aztec C68K/Am-p Professional System \$199

Includes all of the Standard Features

Aztec C68K/Am-d Developer System \$299

Includes all of the Standard and Extended Features

Source Level Debugger \$75

Library Source \$300

Order Now At No Risk Or Call For Information

1-800-221-0440

In NJ or outside the USA call: 201-542-2121

Telex: 4995812MANX Fax: 201-542-8386



Manx Software Systems • One Industrial Way • Eatontown, NJ 07724

SEE FOR YOURSELF WHAT ALL THE FUSS IS ABOUT ...

Order our Manx Aztec C68k/Am with 30-day satisfaction guarantee. We're convinced that once you see SDB at work, no other debugger will ever be good enough again. But if you don't believe us—try us! We're offering an SDB Demonstration Disk for just \$5. Simply call 1-800-221-0440 (NJ call 201-542-2121) and order your copy of Aztec C or your Demo Disk today.

introducing...

EXTEND™

an AmigaBASIC extension

EXTEND is a portable Library of 30 new AmigaBASIC commands that bring the Pizzaz of INTUITION into your AmigaBASIC programs

TRUE INTUITION REQUESTERS and GADGETS

- * Point & click on directory requester gadgets for ease in loading and saving files
- * String and boolean gadget implementation with polling support
- * Custom string requesters

MENU CONTROL

- * Complete menu attribute control
- * Submenu definition
- * Assignment of command key functions

SPEED

- * Written in 100% assembler
- * Compatible with all tested compilers

EASE of USE

- * No program overhead
- * Invoked with a simple LIBRARY statement

Extend Your Horizons for only \$59.95

ALSO AVAILABLE

VIDEO CATALOGER	* Organize your videotape collection	
HOME INVENTORY	* In the event of loss could you list everything you own?	\$34.95 each
MAIL ME	* Manage lists. Make labels. customer lists, greeting card lists, birthdays etc.	

VISA / MC Accepted
Dealer Inquiries and
Phone Orders
Welcome

SUNSMILE SOFTWARE
533 Fargo Ave. Buffalo NY. 14213
716 / 885-5670

SPECIAL INTRODUCTORY OFFER
EXTEND + any other
ONLY \$79.00

SATISFACTION GUARANTEED
Compatible with all
Amiga models

My method for the DRP stock entry may not follow the book's, but it does save memory and keeps entries to a minimum, while eliminating a jumble of numbers. Enter the first DRP purchase manually (A6-H6) from the company statement. Begin using formulas on the second entry. You'll have to be quick to catch VIP's lightning-fast automatic calculations on Amount Invested, Total Shares, and Average Cost per Share.

MACROS, Anyone?

Until you feel comfortable with VIP, forget the macro section. Once you get in the swing with the spreadsheet format, you'll want to set up some single key command macros. These little goodies can automatically read

and perform a "record" of the key-strokes you assign to them. VIP comes with some pre-set shorthand (/P, /FS, /FR, etc.) used in the spreadsheet program. They are easy to remember (first letter of word) and save time. Now it's your turn to create your first macro. First, save and exit the current worksheet. With the cursor at A1:, type your name followed by a tilde (~).

Joe Smith~

Make sure the cell below A1 is EMPTY. This ends the macro. Now you must name the macro. With the cursor on cell A1, which contains the macro, perform the following steps to give the cell a special range name:

From the Worksheet Menu, display Range Name Create (/RNC). At the "Enter name:" prompt, type: \a and

hit RETURN. Hit RETURN again to confirm that cell A1 is the name: \a. Save the macro with File Save (/FS). To execute the macro, move the cursor to an empty cell, hold down the right ALT key and press "a." Congratulations!

Macros can perform both simple and complicated tasks, but they can't read a value entry. All macros must be made into labels. Create a library of these single keystroke commands for use in portfolio management spreadsheets (setting column widths, entering ticker symbols, menus or executing any entry that is often repeated).

Using and creating macros is a subject of its own. An excellent book—listed in Appendix B of the VIP User Handbook—is *The Hidden Power of Lotus 1-2-3: Using Macros*. If you can only afford one reference book, my advice is to buy *Lotus 1-2-3 Self-taught on the IBM PC* by Ira Krakow, Brady Communications Co. For ready-to-use models, *1-2-3 Run* by Lauren and Robert Flast is good. All three books are easy to read and understand.

After you complete and save your stock portfolio spreadsheet, try using the graphing capabilities of VIP Professional. The visual impact of graphing your calculations may not be necessary for personal investment management, but it certainly is helpful. The manual has good examples and text to guide you on your way to completing your stock portfolio with that Amiga VIP touch!

•AC•



AT LAST!

Now Shipping.

To order call toll-free anytime:

Nationwide: 800-452-4445, ext. 1156

California: 800-6269541, ext. 1156

For more information, contact:

α^2

A-Squared Distributions Inc.

6114 La Salle Avenue, Suite 326

Oakland, California 94611

415-339-0339

The AMICUS NETWORK

By John Foust

Fall COMDEX Report

In short, Fall COMDEX in Las Vegas was flooded. It rained for the first few days of the show, flooding the streets of the city. Three inches of water flooded the COMDEX registration tent on Monday.

COMDEX was also flooded with people. A record number attended, topping 90,000. The city didn't have enough cabs or hotel rooms to go around. In a small way, there was also a flood of Amiga hardware and software.

As they did this spring, Commodore hosted a show for its dealers on the Sunday before COMDEX. The featured speaker was Max Toy, Commodore's new president. Toy brings much computer experience to Commodore. He worked for IBM when the PC was launched; he moved to Compaq, a leading maker of PC compatible computers; then he moved to ITT's computer division.

The dealer show brought the first official acknowledgement of several long-rumored products from Commodore. These include a 68020 board for the 2000, a high resolution display with approximately 800 by 1000 pixels in monochrome, and a professional video board for the 2000. More on these later. Commodore also revealed a BridgeBoard based on the 80286 microprocessor that turns the 2000 into an AT clone. No shipping dates were given.

Software and hardware developers had a separate room for showing their products to dealers in an atmosphere a

little less hectic than the COMDEX show floor. This was nice because of the limited space in the Commodore booth. Every developer had the chance to participate in the pre-show dealer presentation.

The developer room had dozens of Amiga 500s and 2000s, and only one or two Amiga 1000s. Following the dealer show, Commodore hosted a cocktail party. Mimetics' Jeff Burger provided live music for the party using SoundScape and a full rack of MIDI instruments.

COMDEX

In the West Hall of the COMDEX show, Commodore had a nice, new booth, topped off with the new Amiga logo on an illuminated sign hanging from the ceiling. The sign was very prominent, visible from all areas of the West Hall. Only the Amiga and PC product lines were on display. The booth had wide open spaces, retreating from the "pack the booth" technique used by Commodore and Atari at previous shows. This approach gave the booth a more professional look.

At one corner, the booth featured a continuous viewing of two video tapes promoting the Amiga. Commodore gave away Amiga baseball caps and "Only Amiga Makes It Possible" buttons. The video tapes included segments filmed at the New York AmiExpo, mixed with shots of the musicians and vocalists in the studio recording the "Only Amiga Makes It Possible" jingle. Commodore appar-

ently intends to advertise, too; they even had a billboard on a street in Las Vegas.

Commodore Products

A new high resolution monitor was shown, part number A2024. No word was given on when this monitor will be available, but its specifications are impressive. Some sources indicated the monitor may be priced in the \$500 to \$600 range. If this is true, it would make the Amiga 2000 a very affordable desktop publishing system. This product makes the most sense when coupled with rumors of a Commodore laser printer.

The monitor's display is 1000 by 800 pixels, by two bitplanes deep. Four shades of gray are possible with no potential for color in this mode. The signal comes from the standard RGB connector on the back of the Amiga and works on all Amiga models.

MicroSearch had the enviable job of demonstrating the new monitor. They showed their City Desk desktop product working in conjunction with an HP Laser Jet laser printer.

The new monitor system was developed by Hedley Davis of Commodore West Chester, working closely with Dale Luck, based in the West Coast Amiga offices. Luck wrote revisions to the operating system graphics software, so the higher output resolution would not be apparent to both the operating system and other Amiga programs.

(continued on page 20)

2090-

Don't miss the boat...



with Amiga expansion products

that limit expansion

SubSystem™

A2000

The Advantage™

A two megabyte RAM expansion card for use in the Amiga 2000 and the SubSystem.

- No-wait-state design
- Auto configures with all Amiga 1.2 software
- Designed to A2000 form factor
- Very low power consumption
- User may install inexpensive 256Kx1 dynamic RAM

Unpopulated (0K)

\$199

Populated (.5, 1 or 2 MEG)

CALL

SubSystem™

A2000

OverDrive™

Direct Memory Access (DMA) SCSI Interface. Just because you have an Amiga 500 doesn't mean that you don't want the speed of DMA. Using the SubSystem with our A2000 card gives you what others only offer to A2000 owners. No matter what Amiga you own, Pacific Peripherals makes a SCSI for you. Our SCSI offers you compatibility with proven Apple Macintosh external storage devices. As a matter of fact, all of our drives are Macintosh compatible. If you use the SCSI in your A2000 you have an additional bonus...the ability to add a hard drive **inside** your Amiga and still use external devices. In addition to 30 megabyte and 50 megabyte (and larger) drives, Pacific Peripherals offers the Infinity removable media drive. Once you have purchased the Infinity, you have **unlimited capacity**. Each 10 megabytes of memory costs a whopping \$18. (Does 100 megabytes for \$180 sound more impressive?) With all this capacity you still get 75ms access time.

OverDrive card only

\$249

OverDrive "hardcard"

CALL

External Hard Drives

CALL

A500

A1000

SubSystem™

Use cards designed for the A2000 with your A500/1000, not out-of-date A1000 cards. The SubSystem gives you two expansion slots for A2000 cards and a space for an optional second floppy drive. The SubSystem fits under your Amiga, completely out of the way. Only 1.5 inches tall, the SubSystem raises the keyboard to the height of an average typewriter and actually makes it easier to use. A UL/CSA-approved power supply is included that guarantees additional cards will not overtax your Amiga. The optional floppy drive is state-of-the-art CMOS design with extremely low power requirements. Cards and disk drive can be easily installed at a later date.

\$249

\$399

with floppy drive

A1000

New!

SubSystem 1000 FOR AMIGA 1000

(415) 651-1905

Pacific Peripherals

P.O. Box 14575
Fremont, CA 94539

Policy: Add 3% for VISA or Mastercard. Allow 3 weeks for checks to clear. Send cashiers checks or money orders for faster shipment. California residents add 7% for sales tax. No charge for UPS ground delivery. Next day and 2nd day delivery available. Prices subject to change.

Infinity is a trademark of Peripheral Land. Macintosh is a trademark of Apple Computer Inc. Amiga is a trademark of Commodore Business Machines. Cage II, the Advantage, SubSystem are trademarks of Pacific Peripherals.



Commodore has also produced a genlock for the 2000 based on the design of the genlock for the Amiga 1000. It should be slightly less expensive because it is a card, not a card in a case. According to a Commodore representative, the announced medium-persistence monitor should be shipping in January.

No More Interlace

Microway showed FlickerFixer, a deinterlacer board for the Amiga 2000. It eliminates interlace flicker by buffering the video image in its on-board memory. It refreshes the screen at the same rate, but the interlace is gone. It also eliminates the dark gap between scan lines.

Microway is well-known in the IBM community for making hardware and software for floating point coprocessor chips. They had their own booth as well. Microway is the first big player entering the Amiga hardware market, akin to the entry of WordPerfect in the Amiga software market.

FlickerFixer can display all 4096 colors. The board will retail for \$595, and they expect to ship in January. It requires a Multisync-type monitor, which is currently very popular in the IBM market. A Multisync monitor sells for about \$575 via mail order, so the total system price is about \$1000.

Taurus Software showed a CAD program called X-CAD. They showed it on the machine running FlickerFixer. It has fast screen updates, which address most complaints about other Amiga CAD programs. Taurus plans future extensions for vector-based font design for laser printers, and modules for doing three-dimensional drawings.

Sculpt 3D was demonstrated on an Amiga 2000 equipped with Commodore's 68020 card. The speed of the program increased quite noticeably. Byte by Byte will ship the successor to Sculpt 3D, called Animate

3D, in mid-December. They also showed the Byte Box, an external memory expansion for the Amiga 500 which holds up to two megabytes of memory.

The Commodore 68020 board will quadruple the speed of some programs. The board has an on-board memory-management unit to oversee as much as an extra two megs of 32-bit memory. The extra memory could be used to bring simple virtual memory to the Amiga. There is also a slot for a 68881 chip. The board runs at 14.2 Mhz, and the 68881 can run at any speed, depending on an add-on crystal.

Commodore also showed the NYIT digitizer and genlock board for the Amiga 2000. Many of its hardware abilities are under software control, so

"The Commodore 68020 board will quadruple the speed of some programs. The board has an on-board memory-management unit to oversee as much as an extra two megs of 32-bit memory."

you can switch between two video sources and adjust the genlock from sliders on the control software. The digitized images are stored in local memory until copied to the Amiga, if you want to convert them to IFF. Unfortunately, according to some Commodore forecasters, this board is still several months from shipping and is currently undergoing FCC testing.

80286 Board

The announced 80286 BridgeBoard was held up at customs for several days. It was developed in West Germany, and Commodore West Chester apparently does not have a working prototype.

Unfortunately, I left COMDEX before this board was shown. The 80286 is the microprocessor used in IBM AT

machines. It runs faster than the 8088 chip used in the BridgeBoard now shipping for the Amiga 2000. The Commodore 80286 board reportedly runs at 6 Mhz, which is slow in today's world of 8, 10, and 12 Mhz AT clones. It will certainly be faster than the 4.77 Mhz BridgeBoard, but real-world use will be the only indicator of relative speed. Differences in video display speeds may be the critical links.

Although the 80286 card is faster than today's 8088-based BridgeBoard, both cards must interface with the proprietary video display software in the Commodore BridgeBoard design. The update speed of this software will be very important to users seeking faster PC compatibility in an Amiga 2000.

Programs like word processors expect to write to the screen at faster rates. These programs may appear sluggish when run on the BridgeBoard system. To be certain, the BridgeBoard hardware and software will never be as fast as the dedicated video hardware in a real PC clone. Of course, users can always implement a PC video card and a separate monitor with the Amiga 2000. If they want full video speeds. The Amiga 2000 PC compatibility is also currently limited to IBM monochrome and CGA graphics resolutions. If you want EGA or VGA resolutions, you must use an external monitor and separate video card.

After COMDEX, representatives of Commodore at a user group meeting said there may also be an XT-type BridgeBoard, with a 8088-type chip running at 8 Mhz.

Video Toaster

NewTek showed the "Video Toaster" at COMDEX for the first time. The Video Toaster is a real-time, full color digitizer and genlock, with digital special effects abilities. It is a prototype of a device for sale next year. Flyers at the show said it would sell for \$799.

(continued on page 22)



Other Products From The Other Guys

Reason	\$395.00
Omega File	\$79.99
Promise	\$49.99
KEEP-Trak GL	\$49.99
AMT (Amortization Program)	\$39.99
Match-It	\$39.99
Math-A-Magician	\$39.99
Talking Story Book (Christmas Stories)	\$39.99
Musical Slide Show Demo	\$ 5.00

**Call or write for
more information.**

SYNTHIA High Performance Digital Synthesizer

A state of the art music tool which will:

Create digital IFF Instruments for use with nearly all music programs!

Modifying existing IFF Instruments. Use **SYNTHIA** on digitized samples to add reverb, wow, and other enhancements.

SOMETHING FOR EVERYONE:

Additive Synthesis - a traditional method which can create almost any type of instrument.

Plucked String Synthesis - simulates plucked strings . . . right down to the 'pluck'.

Interpolative Synthesis - a method which introduces the natural imperfections found in instruments.
(Instruments such as brass, woodwinds, pianos, etc.)

Percussion - build your own drum set . . . create any drum you desire.

Subtractive Synthesis - a simple method of creating instruments.

Special Effects - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .

IFF Music Player - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

IS IT LIVE . . . OR IS IT SYNTHIA?

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!

Why buy digitized instruments when you can **SYNTHIAsize** them?

Requires AMIGA 512K

\$99.99

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga



THE OTHER GUYS

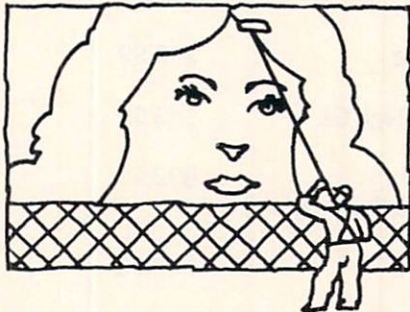
55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
(800) 942-9402



NOW SHIPPING

GET THE BIG PICTURE



"THE BIG PICTURE"

Print your IFF/HAM pictures in full color up to 104 feet x 104 feet. "Real" multiprocessing and overscan printed because printing is done from the file not the screen.

\$29.95 for the following printers in one package:

Canon PJO80A / Okidata 20 color Radio Shack Quadran Quadjet / Epson FX-80, FX-100, EX-800, EX-1000 / Fujitsu 1200, 2300, 2400 NEC P6, P7, CP6, CP7 / XEROX 4020

ATI-LINK

An IFF/HAM toTARGA file conversion program only \$100.00

Preferences Drives

Quadran Quadjet \$30.00 NEC CP7 \$50.00

Big Picture is a trademark of Lightning Publishing. Amiga is a registered trademark of Commodore Ltd. Targa is a registered trademark of AT & T

Lightning Publishing

1821 N. Ohio St., Arlington VA 22205
(703) 534-8030

Name _____		
Address _____		
City _____	State _____	Zip _____
Amount _____	Program _____	

The Toaster was demonstrated at COMDEX after I left Las Vegas. I saw the Toaster in action at the FAUG user group meeting the week after COMDEX. The color video camera was pointed at audience members R. J. Mical and Leo Schwab, and special effects manipulations were performed in real time. The Toaster put sixteen smaller copies of the image on the screen at once, moved a smaller box around the screen, flipped the box end over end, mapped the flat image to a round surface, and "pixelized" the image, all in real time. It is very impressive, but it isn't shipping yet.

University of Lowell's Center for Productivity Enhancement showed their parallel processing board for the Amiga 2000. It demonstrated edge detection at several frames per second from a live video source. It uses NEC image processing chips that work in parallel. This means that for more speed, you just add more chips. They estimate a price of about \$2000.

Changes

Several companies have made changes to their products. Finally Technologies showed their Hurricane Accelerator board, a 14 Mhz 68020 speed-up card. The list price has increased to \$595 since the Commodore Show. Gold Disk showed Professional Page again and said the program would be out in December.

Mimetics has changed the name of their genlock from Imagen to AmiGen. The retail price is \$179.95, and Mimetics said it would ship the week after the show. Roger Powell's MIDI program, Texture, has been "Amigaized," so it looks less like an IBM PC program. The list price is \$199.

WordPerfect Corporation reported tremendous sales of Amiga WordPerfect. They plan to convert other products to the Amiga. They reportedly recouped their Amiga porting investment in the first month of sales.

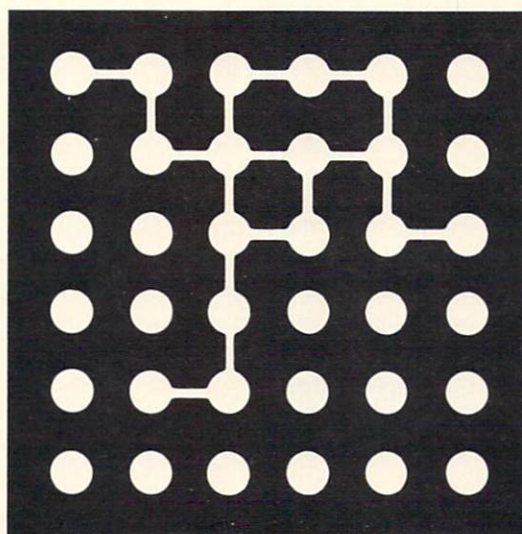
Xerox is still using the Amiga to demonstrate its color printers. Hewlett-Packard supplied Commodore with several HP PaintJet printers for the booth. Both HP and Xerox were using preliminary versions of the AmigaDOS 1.3 printer drivers. The new drivers bring as much as a six-fold increase in printing time over the AmigaDOS 1.2 printer drivers.

Progressive Peripherals showed a slew of new Amiga products. They had their own booth, apart from Commodore. They showed a prototype real-time video digitizer, scheduled to sell for less than \$500. It digitizes to its own memory, then downloads the data to the Amiga through the parallel port. Other hardware plans include a genlock and various products for the Amiga 500, including several memory boards, an external drive that is much smaller than the Commodore drive, and a 68020/68881 enhancement.

Progressive is not short on software, either. PIXmate is an image processing program for manipulating IFF images. It converts between resolutions, as well as accomplishing more advanced techniques, such as edge detection and image enhancement. PIXmate is very fast because it uses the blitter for many operations. Progressive also showed demos of two CAD programs, an accounting program, a disk management program, and a program to let you use Commodore 64 peripherals on the Amiga. All these appeared far from completion.

Superbase Professional will reportedly be available before the end of the year. This one looks like it has more kitchen-sink features than any other program. It has programmable telecommunication features for moving data to other computers, as well as built-in word processing and mail merge facilities.

•AC•



What's a Haicalc?

(pronounced hī-călc)

Haicalc™ is the best valued spreadsheet available for the Amiga. If you have ever had to project budgets, calculate expenses, or prepare a financial statement you need Haicalc. For only \$59.95 you receive a powerful program that fully utilizes the Amiga Intuition user interface.

You have the convenience of simple point & click selection of files under Workbench, pull down menu selections for commands, and economical use of your computer's memory -- program size is only 120k. You harness the power of multi-tasking and macros with

a maximum spreadsheet size of 9,000 x 9,000. Compare this to the competition. Haicalc is a powerful solution for a painless price.

SPECIFICATIONS

Maximum Spreadsheet Size — 9,000 rows by 9,000 columns*

Number Precision — Accurate to 8 digits

Interlace Display Ability — Up to 44 rows of data per window

Window Resolution — 1,000 by 1,000 pixels

Address Space — 18,000,000 cells*

Program Size — 120k

Sparse Matrix Allocation — Uses memory only when a cell contains data

Copy Protection — None

System Requirements — 512k Amiga, 1.2 Kickstart, Single floppy drive

*All capacities are subject to limits of system memory and storage.



HAITE X

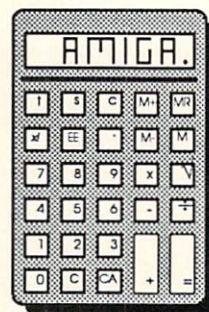
HAITE X RESOURCES

208 Carrollton Park
Suite 1207
Carrollton, Texas 75006
(214) 241-8030

Money Mentor Version 2.0

A Personal Financial System

by Stephen Kemp



A home accounting system should do more than just replace your checkbook. After all, who wants to do the same work twice? In addition to keeping your checkbook balanced, a home accounting system should do the following things:

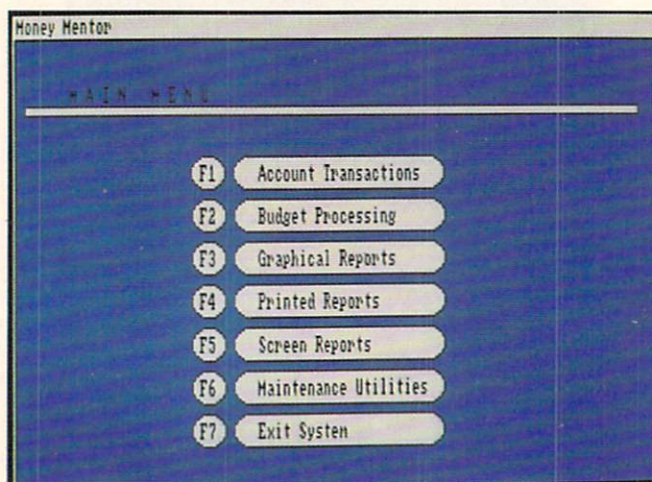
1. Keep track of other types of accounts (savings, IRAs, and credit cards).
2. Allow you to make budget items for things like your salary, groceries, rent, car payments, and mortgages.
3. Keep track of day-to-day transactions affecting these accounts and budgets.
4. Provide a clear and meaningful way to compare your actual performance to your projected budgets.
5. Print checks.
6. Allocate transactions to multiple budget categories.
7. Provide query capabilities for your accounts.
8. Simplify entering transactions that recur from month to month.

Money Mentor is a home accounting system by Sedona Software. With Money Mentor version 2.0, you can create accounts and budgets, enter transactions against your accounts, transfer funds between accounts, print

checks, and produce graphs and reports to keep track of your assets. Essentially, Money Mentor does everything in the above list, and more.

Getting Started

The documentation is quite good. It uses understandable examples and is



divided into sections corresponding to the menu choices in the program. This makes it easy to refer to the documentation for help while you are using the program.

The first few chapters explain the basic concepts of home accounting and how Money Mentor can work for you. A tutorial begins in Chapter Four. Although some people may be able to use Money Mentor right out of the package, most will find the tutorial valuable.

You must create a data disk to begin using Money Mentor. Money Mentor makes this disk if you select the "Create Data Disk" icon. Once you have a data disk, selecting the "Checks" icon starts Money Mentor. If the data disk is not in a drive, Money Mentor prompts you to place the disk in any drive. If you have only one drive, have no fear. Once you place the data disk in the drive, the program disk is no longer required. This means you won't spend your time swapping disks.

Money Mentor does an excellent job of using both the keyboard and mouse. After using the system for a while, you almost never need to touch the keyboard. For those of us who are not mouse fanatics, the keyboard does everything the mouse can do. Money Mentor is menu driven, and functions are selected with the mouse or the function keys. Once you arrive at the desired function, Money Mentor accepts data through Scrolls.

Scroll It In

All data is entered through Scrolls. A Scroll is a unique data-entry field with which you can not only type the required information for a field, but you can also select items from a scrollable area underneath the field. An item can be selected from the Scroll with the mouse or arrow keys.

As an added feature, if you type the first letter or first few letters of an item in the Scroll, and then hit an arrow key, Money Mentor places the next item that begins with those letters in the selection box. This feature is a quick way to search through large lists of items.

Money Mentor can also create "Smart Scrolls." A Smart Scroll knows the past relationships between Accounts and Budgets. Many of your transactions (car payments, mortgages, and electric bill) are similar each month.

Money Mentor can "Study" your accounts, so the next time you begin to enter one of these transactions, a default amount, budget label, tax status, and note are placed in the Scroll under your field. Of course, you can easily override the default easily by typing the correct information or choosing a different item from the Scroll. Sedona claims their Smart Scrolls save 70% of the required typing for an entry. After using it myself, I must agree.

Defining Accounts And Budgets

At the top of everything are the Accounts. Money Mentor defines Accounts as "the buckets in which Transactions occur." Money Mentor maintains up to 30 different accounts. To create an Account, you must give it a name, define what type of account it is, and enter its starting balance. Accounts can be any one of the following types: Checking, Saving, Cash, or Bank Card. The default account type is Checking.

Once Accounts are set up, you are ready to create Budget items. Money Mentor keeps track of up to 100 income Budget items and 100 expense Budget items. You should make budget items for your salary, groceries, mortgage, car payment, cable TV,

want to compare budget to actual performance, but entering data as you create the budgets makes the most sense. With good budgeting, you will know exactly where your pennies are going.

The screen into which budget amounts are entered is very similar to a spreadsheet screen. There is room for thirteen Budget items and Monthly Totals on the vertical axis. Five Months with a Yearly Total are on the horizontal axis of this screen. What you see is not all you get, though. The other months can be scrolled onto the screen using the left and right arrow keys; if you have more than thirteen Budget labels, they can be scrolled onto the screen using the up and down keys. The Totals always remain on the screen.

A red box in the table indicates where you enter data. With the mouse or cursor keys, you can move the red box wherever you want. Data is entered with a Scroll and is placed in the space occupied by the red box. You can also REPEAT a value through to the end of the year, or COPY a value from one space into another. It can't

get much easier than that. While amounts are entered, the totals, by Month and Budget item, are calculated automatically and displayed in the appropriate places.

Transactions

After you establish your Accounts and Budgets, you enter Transactions.

A Transaction shows where money enters and exits in your accounts and budgets. You'll spend most of your time entering Transac-

Money Mentor

Ledger for The Club Checking

Number	Date	Name	Amount \$	Budget	Note
118	2/28/87	T S Oil Co	\$ 58.00	GMC Car	Gas
117	2/28/87	G E	\$ 200.00	Savings	Monthly
120	2/28/87	Tophs House	\$ 100.00	Medical	Dental
121	10/31/87	S And P National	\$ 100.00	Savings	Monthly
6	11/12/87	Smith Company	\$ 3,000.00	Salary	Monthly

Enter Next Action to Be Taken:	Next Action:	Number	Tax	Month	Cleared
Edit	Deposits	118	Yes	Feb	No
Checks		117	Yes	Feb	No
Deposits		120	Yes	Feb	No
Done	Balance	121	Yes	Oct	No
Pay Back	\$ 2,900.00	6	Yes	Nov	No

dividends, electricity, etc. A Budget item must be given a label (name) and a tax status (taxable or deductible). You can also define the tax status as "Ask Each Time," meaning you can make that decision as you enter transactions.

Money Mentor

Create Expense Budget

Expense	Jan	Feb	Mar	Apr	May	Year Total
Club Checking	2,000.00	2,000.00	2,000.00	2,000.00	2,000.00	24,000.00
Fica	46.50	46.50	46.50	46.50	46.50	558.00
GMC Car Payment	200.00	200.00	200.00	200.00	200.00	2,400.00
Groceries	90.00	90.00	90.00	90.00	90.00	1,079.10
Household	66.00	66.00	66.00	66.00	66.00	792.00
MasterCard	35.00	35.00	35.00	35.00	35.00	420.00
Medical	25.00	25.00	25.00	25.00	25.00	300.00
Mortgage	700.00	700.00	700.00	700.00	700.00	8,400.00
Other	100.00	100.00	100.00	100.00	100.00	1,200.00
Savings	500.00	500.00	500.00	500.00	500.00	6,000.00
Software Store	35.00	35.00	35.00	35.00	35.00	420.00
Taxes	110.00	110.00	110.00	110.00	110.00	1,320.00
VISA	36.00	36.00	36.00	36.00	36.00	432.00
Month Total	4,003.50	4,013.50	4,023.50	4,038.50	4,023.50	48,267.00
Value	700.00	F1 Edit	F2 Copy	F3 Repeat	F4 Exit	

Entering Budget Amounts

Once you establish Budget items, you should enter expected monthly expenditures and receipts. Monthly Budget amounts are not required until you

(continued)

tions, so this is where you want the program to function best. Money Mentor is no Transaction slouch.

Transactions have 6 parts: the NUMBER of the Transaction, the DATE of the Transaction, the NAME of the source/destination of the Transaction, the AMOUNT BUDGET item to which this Transactions should be allocated, and a NOTE documenting the Transaction.

The Number of the Transaction is entered first. If you are working with a checking account, and you are writing a check, this entry should be the check number. Once you establish the first number for this type of Transaction, Money Mentor increments the number for succeeding Transactions as the default.

Money Mentor defaults the Date field to the date entered at startup. You can change the date by typing a new one or selecting one from the Scroll.

The Name field is where you record the source or destination of the transaction. If this is a check, type the name of the person to whom the check is being written. Later, if you choose the "Print Check" option, the Name entry is written in the "Pay to" line of the check.

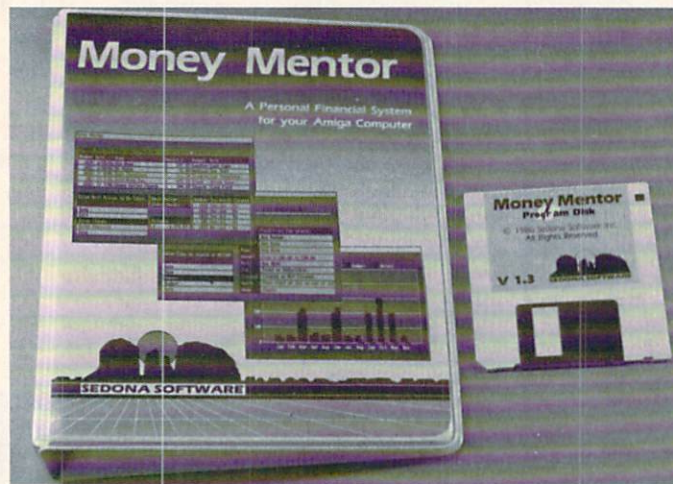
The Amount field is self-explanatory.

You can allocate this Transaction to a Budget item by selecting or typing the appropriate Budget item's label.

Again, the scrollable menu underneath the field saves you some typing. If a Transaction is divided between various Budget categories, you can select "Split" from the Scroll.

"Split" is a really handy feature. Let's say you have written a check to your mortgage company for \$750.35, and you want to allocate \$650 of the check to interest, \$50.35 to escrow, and the remainder to principle. First, enter a check to your mortgage company for \$750.35 and select "Split" as the Budget label. Once you finish the check Transaction, a field labeled "Split Balance" holds the \$750.35.

Money Mentor now lets you enter Transactions to settle the split allocation. As you enter Transactions, the "Split Balance" decreases. When this field is \$0, you are finished allocating. Split allocations allow you to make very detailed budgets.



Entering a Note helps document the Transaction. Again, if you are issuing a check, you would normally write in the note field of the check. If you choose the "Print Check" option when the Transaction is complete, your note is printed on the check.

When you finish entering a Transaction, you can do another Transaction, edit a previous one, or print a check for the last Transaction. Yes, you can print checks with Money Mentor. An order form is available in the back of the documentation for the proper type of checks.

Smarter Than The Average Program

After all your Transactions are entered, choose "Study Account" from the menu. This function creates and maintains the Smart Scrolls. You want to do this! Suppose you made a car payment to your bank this month. Next month when you indicate a car payment, Money Mentor defaults to the payment Amount, Budget item, and Note field from last month. Just click the mouse or hit the return key to accept the defaults. You won't believe how simple your repetitive transactions become.

How Are You Doing?

Once you have a few months under your belt, you will want to know how well you are following your budget. Money Mentor provides a number of reports that can be sent to the screen, file, printer, or displayed in a graph (Sometimes a graph is more useful than a printed page of numbers.). Money Mentor provides more than 50 status reports, including one noting how much you vary from Actual to Budget, and another with your Top 20 Problem areas.

When your bank statement or charge card statement arrives, choose "Balance

Account" from the menu. With Money Mentor, to balancing accounts is as simple as indicating which Transactions have cleared and which have not. Money Mentor does the rest. You don't have to use a calculator or scratch on paper.

Inevitably, you will discover a check that was entered wrong, or you will want to know whether the cable bill was paid last month, or you will want to know how much the bank has charged for holding your money. You can find out these things with Money

Mentor's search capability. You can search by any of the Transaction fields, by Tax category, by whether it has cleared or not, or by month. Money Mentor lets you combine criteria for the search, so you can be as precise as necessary. Once you find a transaction matching the search criteria, you can edit it, continue the search, or print the Transaction.

Wishful Thinking

Money Mentor has plenty of strengths. However, there are a few things on my wish list. Nothing on my list should prevent anyone from using Money Mentor, though. On the contrary, Money Mentor is an excellent product; these are just things I think would make it even better.

It would be nice to set up automatic accounts. These are accounts that are automatically entered on or after a specific date. This feature could track automatic withdrawals (without a check) from your checking account on a certain day each month.

A calendar/reminder for entering important dates (such as when fees to organizations are due) would be useful. Many people need to be reminded of birthdays and anniversaries...

Money Mentor has a limit of 100 income Budgets and 100 expense Budgets. For most people, this will be more than enough. If you start making very detailed budgets, however, this may become a problem. It might be better if you could determine how the maximum 200 budget items should be divided because the expense Budget items almost always greatly outnumber the income Budget items.

When entering split allocations, you are limited to the same type of transaction for settling the split. This means that you can't enter a deposit for \$2000 in your checking account and split the allocation to salary for

\$3000 and \$1000 taxes. The only way I found to mix income and expense allocations was to make a separate salary account and transfer \$2000 to my checking account.

Finally, it would be nice if Money Mentor supported more than one check type. It currently supports a two-part check. Support for smaller, single-part personal checks would be a bonus.

Summing It Up

Money Mentor will do more than just replace your checkbook. It will help you make budgets, track expenditures, and create financial plans. With Money Mentor, you can keep good records very easily. It makes excellent use of the keyboard and mouse. The menus and documentation are fairly easy to follow, and the data screens are sensible. Money Mentor's Smart

Scrolls, which are probably the product's best feature, will save you a lot of typing time.

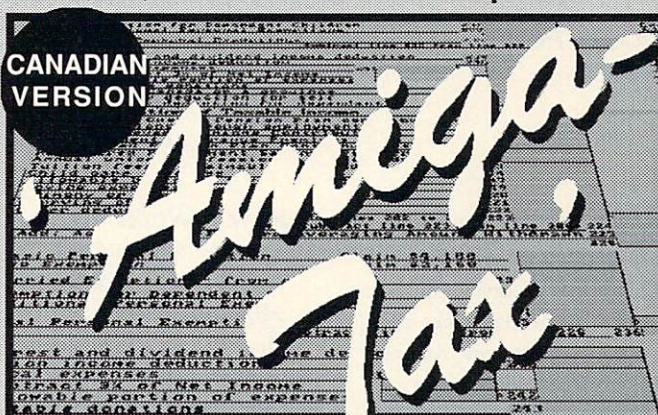
If you are ready to keep track of your hard earned money, Money Mentor can certainly help you.

•AC•

Amiga-Tax II

Tax Planning & Preparation Software

Pay **\$69.95** once then receive
Yearly Updates for **ONLY \$29^{.95}**



- All Schedules, All provinces, Gov't Approved
- PRINTS tax returns No manual transcribing!
- Save Time, Money, and Aggravation
- Useful all year for tax planning

Send \$10 for Mini Version & Save \$10 on final cost.

Datamax Research corp
Box 5000, Bradford, Ontario L3Z 2A6

Money Mentor
list price \$99.00

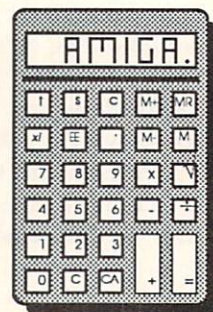
Sedona Software

11844 Rancho Bernardo Road, Ste. 20
San Diego, CA 92128
(619) 451-0151

Investor's Advantage

A Market Analysis Package for Charting Technical Indicators

by Richard Knepper
People Link: Stryker



In an age of increasing market volatility, investors need superior stock selection techniques in order to create portfolios which minimize risks and maximize returns. Software Advantage Consulting Corporation has written a program which uses technical indicators to select stocks which have performed better than the market average. The program also provides information to help investors time their buy and sell decisions to take advantage of market movements.

Investor's Advantage allows you easy access to stock quotes through the Warner Computer Systems Stock Market History Data Base. The program can automatically dial up Warner and download the information you specify.

How It Works

Creating and maintaining files for the equities you want to track is relatively painless. Selecting "Create a New History File" from the main menu brings up a screen which allows you to specify the ticker symbol, the frequency of updating, the length of the moving averages, the P/E multiple, and whether the file is for an equity or an indicator.

Once you have created a file, you can add data to it, either manually or automatically. Manual addition is accomplished by selecting "Add to Existing History File" and entering the stock data required. In order to do this, you must have access to a financial publication.

Automatic updating requires only a Hayes Compatible modem. A control program allows you to specify the equities and indicators you wish to update. (The control program also controls access to Telenet.) You must only tell the program which days to download. The entire process is extremely quick and easy.

Maintaining your history files is quick and easy, too. When using a computer to help make investment decisions, it is essential to double-check the data the computer is using to make its decisions. Investor's Advantage allows you to view and edit the data in a file, as well as delete the entire file.

(continued on page 30)

A Poor Man's Guide to the Stock Market

In today's volatile stock markets, timely and accurate information is vital. In order to assure profitable investments, the investor must be able to gather all available information and act on it immediately. Because this completeness is nearly impossible, most investors devise a system that allows them to make reasonable decisions based on incomplete information.

Two major schools of thought have developed. The first, the Fundamental approach, studies the relationship

between risk and return in the market. This relationship equates supply and demand to yield a market price. The second school of thought, the Technical approach, is not concerned with background data. Instead, it concentrates on the movement of the market itself. *Investor's Advantage* is Technical.

Before you buy *Investor's Advantage*, you should get a feel for the program's theory. Fundamental and Technical investment strategies are drastically different, and there are

arguments both for and against each. If you are going to use *Investor's Advantage*, you should know the limitations of its techniques and feel comfortable with them.

The Fundamental approach has been widely acclaimed by the academic community. The major premise of this approach is that investors are only willing to buy a stock if their return—in the form of dividends and capital appreciation—is commensurate to the risk they face. If a stock does not compensate the investor fairly for the

risk he faces, he takes his money elsewhere. This lowers the demand for the stock, and thus lower the price, so it provides an equitable risk/return relationship.

The net result of this money movement is that the market prices all equities fairly. Investors can then build portfolios that reduce the risk of having all their eggs in one basket. (An example of this risk is investing in one company. The company may be doing fine one day and lose everything in a \$12 billion lawsuit the next.) These portfolios can also be structured to provide specific levels of return, as a percentage of market return, both positive and negative.

There are two major arguments against Fundamental analysis. The most important is that it is too difficult to determine the exact level of risk in a specific equity. Historical risk is usually used, but this is only an approximation. The second argument is that the risk/return relationship is not always repeatable when external market conditions have been the same. This argument is less pervasive and only indicates investors are not using complete information.

Technical analysis is the system most often used by professional traders. They believe the fundamentals affecting a given equity cannot be known soon enough to act profitably. They then conclude that unknown changes in the fundamentals are reflected in the price action of the equity. The analyst believes price changes form patterns similar to past patterns.

Technical analysis can be subdivided into major disciplines. The first is charting, which attempts to take advantage of short term fluctuations in price. The technician must closely follow the market and be willing to take very short term positions. Most true chartists do not take positions lasting more than one day.

The second type of technical analysis is the tracking of technical indicators. This approach is very similar to the first, but is more concerned with longer term movements in the equity. In most cases, this technician collects information either daily or weekly. This time-frame is followed deliberately to eliminate short term volatility. The technician then tries to interpret the information for long term trends.

The major difference between the two technical disciplines is that charting attempts to determine where the market will move in the short term, while tracking technical indicators tries to determine where the market is moving in the long term.

Both methods have pitfalls. Chartists must receive information in a very timely manner. This problem makes charting an unrealistic tool for investors not on the exchange floor. Charting is difficult even if a real-time ticker is available. Most investors should not consider short term charting because they will not be able to properly time their buy/sell decisions, and will usually get in just before the market reverses.

Tracking technical indicators is much more feasible for the average investor. These indicators are more useful for long term investment strategies. They indicate major market movements and help the investor time his entry and exit in the market. The main problem is that trends are not easy to spot. By the time indicators have signaled a clear trend, it may be too late to take advantage of it. This often means the investor gets in too late, buying the market tops and selling the bottoms. He is then doing worse than he would be if he weren't using any information at all.

There are also problems with trading from a "contrarian" standpoint. This theory should be used only after the market has demonstrated a clear trend. Even then, it is difficult to determine

when the reversal will happen. Often, one can begin to fade the market (take a position opposite to the market's movement) just as it begins to build momentum. A number of large Stock Index traders took contrary opinion stances during Black Monday and were looking for new jobs on Tuesday.

Even though there are valid arguments against both fundamental and technical trading, that doesn't mean that they aren't both useful. Each system provides valuable information which can make the difference between success and failure in the market. When I was actively trading futures at the Chicago Board of Trade, I took long term stances, using fundamental analysis as my primary source of information. I tempered these stances with information gained through analysis of technical indicators. I timed my entry into the market through charting.

Not everyone has twelve plus hours per day for market analysis. If you want to be in the market and make your own investment decisions, I suggest following the fundamentals and using the technicals to temper your information. Don't worry about timing your entry exactly. Entry time usually won't make or break a long term position. If you don't have the time to follow the market closely, consider investing in a fund. Funds give you the diversification you need and allow reasonable returns without spending a great deal of time.

If you are serious about investing in the market, Investor's Advantage gives you the information you need in a reasonably timely manner. The price of software and information retrieval is low enough to make it easily affordable for armchair investors. Remember, Investor's Advantage is a tool to help you make informed investment decisions. It is not an end all, nor is it intended to be.

-- Richard Knepper

In summary, the process of creating and maintaining the equity and indicator history files is simple, allowing you to concentrate on the primary purpose of the program: helping you make money.

What Investor's Advantage Tracks

The program uses its historical database to calculate and display a number of different technical indicators.

Moving Average is calculated by taking the price of an equity over a period of time and calculating the average. For example, a thirty week Moving Average is based on the last thirty weeks of data; each week's Moving Average changes when a new week is added and the oldest week is dropped. The Moving Average calculation factors out day-to-day volatility and indicates an overall trend in the equity. The program allows three different Moving Average durations to be specified. Short, intermediate, and long-term trends can be compared.

Price Momentum shows the relative change occurring in the three Moving Averages. This is done by dividing the current price by the oldest price in the Moving Average. Thus, a thirty-week Moving Average Price Momentum is equal to the current price divided by the price of the equity thirty weeks earlier.

High/Low Chart displays the high/low/close, the Relative Strength, and the volume of shares traded. The chart generated is the same type published in newspapers. The most obvious use for this chart is to perform technical analysis. (As explained in "A Poor Man's Guide to the Stock Market," technical analysis is not the same as analysis of technical indicators.)

Relative Strength calculates the change in the Moving Average for all equities and compares them. This

provides a ranking system to indicate those stocks which have demonstrated superior performance. This is useful to the technician who wants to create the best possible portfolio of stocks.

Monthly Percentage Change calculates the percentage change in each equity that has occurred month-to-month for the past thirteen months.

Daily Stock Market Barometer displays five indicators that give insight into the movement of the market as a whole (see Figure 1).

1. DJIA displays the daily movement of the Dow Jones Industrial Average.
2. Advance/Decline Line displays an indicator of the number of stocks that rose in value versus the number of stocks that dropped. This gives an indication of the sentiment of the market.
3. Odd Lot Short Ratio represents a contrarian viewpoint, which assumes that a group of investors are predicting a market movement, and that their predictions will be wrong. If the Odd Lot Short Ratio is increasing, this group is shorting the market (selling stocks), and the market should move upward. If the ratio is declining, they are long (buying stocks), and the market should decline.
4. Put/Call Ratio, another contrarian indicator, reflects the attitude of options buyers, who are, for the most part, speculators. Puts are bets that the market will move lower; Calls are bets that it will move higher. If more speculators are buying Puts than Calls, they are predicting a downturn in the market. A contrarian would then believe the speculators to be incorrect, and expect a market upturn.

5. Overbought/Oversold Ratio is a ten-day composite of the number of stock advances and declines. A high value indicates that the market has been under heavy buying pressure. The

number of future buyers should drop off soon, and the market should decline.

Weekly Stock Market Barometer

displays five indicators which give a more generalized impression of the market movements (see Figure 2).

1. DJIA indicates the weekly movement of the Dow Jones Industrial Average.
2. NYSE indicates the weekly movement of the New York Stock Exchange composite index (a.k.a. Big Board). This lists a greater number of stocks and can give a more accurate indication of the market sentiment.
3. New Highs/New Lows lists the number of stocks which have made new highs or new lows. Following the precept that "what goes up must come down," it is expected that a large number of new highs indicate a market which has reached its peak and will soon decline. High numbers of new lows indicate an oncoming market upswing.
4. Specialist Short Ratio attempts to track the sentiment of the floor specialists. Specialists are floor traders assigned the task of making the market for a particular equity. They are usually better-informed than the populace and can better predict the movement of their stocks. If they are selling stocks, then the Short Ratio will be high and a downturn can be expected.
5. Twenty Most Active Indicator compares the sum of the last three weeks' advances to the sum of the declines. An increase in the indicator shows underlying market strength, while a decrease indicates weakness.

The Manual

The program comes in a green plastic folder about the size of a large paperback book. The manual is about

(continued on page 32)

The Works!

THE AMIGA® COMPUTER STARTER KIT WORDPROCESSOR • SPREADSHEET • DATABASE

Have all of the above instantly at your fingertips on ONE diskette at ONE low price in ONE package. **THE WORKS!** is ONE complete system. Never again will you move from one program to another, forced to learn a new manual with new instructions. In **THE WORKS!** each module uses similar pull down menus and familiar commands. You said "Give me the works". Now in ONE package, you get **THE WORKS!**

Included is a powerful spreadsheet module (Analyze!) that is as useful in the home as in the office. Whether for your personal budget and check register or your company's accounting and forecasting needs, the versatile pull down menus and keyboard shortcuts make this module easy to use. Its *multicolor 3-D graphs, special macro language and compatibility with*

Lotus 1-2-3 make it powerful as well.

The full featured word processing module (Scribble!) includes a spelling checker and mail merge facility. Full support of the Amiga clipboard is provided as well as complete styling control. You may mix **bold**, *italic*, and underline in various combinations on a single line, and show 4 documents on the screen.

The professional database module (Organize!) helps you collect and manage information or data easily. The reports you prepare are completely customized and can be printed to paper, screen or disk. Portions of records may be blocked for confidentiality. From a recipe file to a customer mailing list, all information is at your fingertips.

Developed by:



Micro-Systems Software
Quality Software Since 1979

Scribble!, Analyze!, Organize!
priced separately total \$349.85

Now...The Works!

only **\$199⁹⁵**

See your local dealer or call:

**Brown-Wagh
Publishing**

1-800-451-0900

1-408-395-3838 (in California)

16795 Lark Ave., Suite 210, Los Gatos, CA 95030

Amiga is a registered trademark of Commodore-Amiga, Inc. Lotus 1-2-3 is a trademark of Lotus Development Corporation. MSS Works!, Analyze!, Scribble! and Organize! are trademarks of Micro-Systems Software.

seventy pages long, includes a brief introduction, and explains how to install and run the program.

The manual is very straightforward and is tailored to the novice user. It gives clear and concise instructions explaining each of the program's functions, along with its significance. I found that the manual complemented the program quite well. The only thing missing is an index, but an index isn't absolutely necessary, since the manual is categorized by menu selections.

The manual does not, however, explain the indicators the program tracks. It gives only a brief synopsis of each indicator, similar to the descriptions I included in this article. I certainly wouldn't feel comfortable using this trading system based solely on the information in the manual. Since the program is inexpensive, I recommend that each purchaser also buy a book on technical analysis and indicators. Many good references are available.

Problems and Limitations

Investor's Advantage has been imported from the IBM-PC/XT/AT/Clone/ad nauseum family, and there are a few residual quirks. Not all of them can be blamed on the MS-DOS operating system, but MS-DOS is a good scapegoat. For example, none of the history files you create can have the ".dat" suffix. This file structuring format is not necessary on the Amiga, and should have been removed when the program was ported. Another problem is that history files have a maximum of 229 records—far too few. There is no good reason for this limitation, and many serious investors

will find it a hindrance when tracking equities over long periods of time.

Although I found the program quite easy to use, the onscreen menus were too cluttered. The options on the main screen should have been sepa-

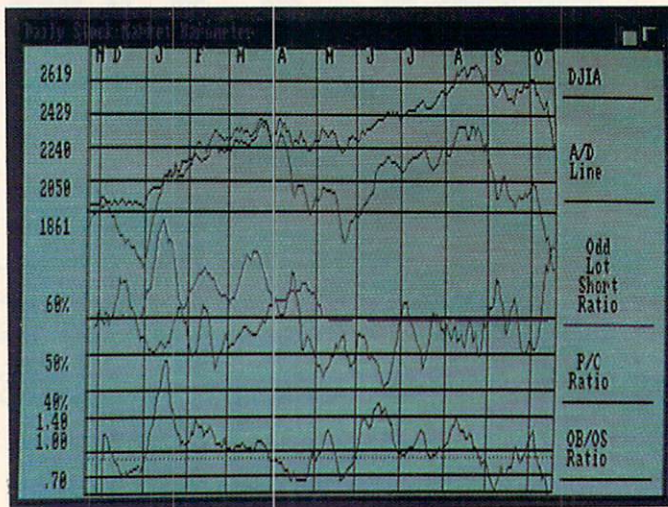


Figure One: Daily Stock Market Barometer

rated into functional groups to make selection easier. In the program's defense, selections *can* be made by choosing the selection number, by pointing at and clicking on the

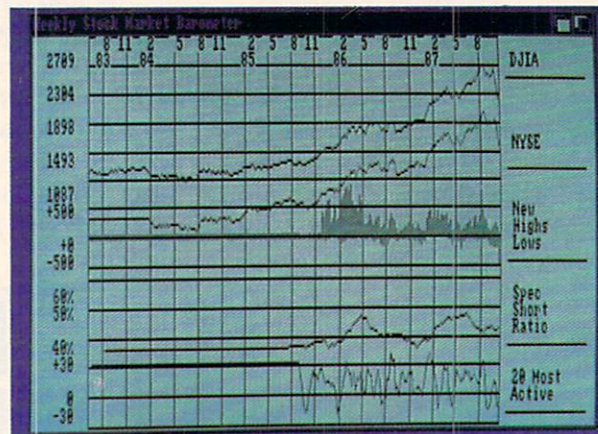


Figure Two: Weekly Stock Market Barometer

selection desired, or by using pull-down menus. Although the pull-down menus are organized functionally, a cluttered screen is still annoying to look at.

The graphs may also appear a bit cluttered at first, but this is due to the amount of information being pre-

sented. Often a single graph shows a number of indicators, each with its own vertical axis. This can be confusing, until you become familiar with the graphs.

Another problem is that the program uses standard Workbench colors, which means when graphs are printed (They are printed in inverse mode.), the background is slightly gray. This is not only annoying but also wears out ribbons quickly. There is no Preferences on the disk, so the system-configuration file must be changed from another Workbench and placed on the disk, either by direct copying or by using Preferences with DEVS: assigned to the Investor's Advantage disk.

The program is slowed considerably by disk access. The menus are organized by indicator, so whenever a menu selection is chosen all stocks must be reloaded from disk. Tracking all of the indicators for a particular stock can be time-consuming. It would have been much simpler to allow you to select a stock and then have the program chart its technical

indicators, rather than make you select the stock after choosing the indicators.

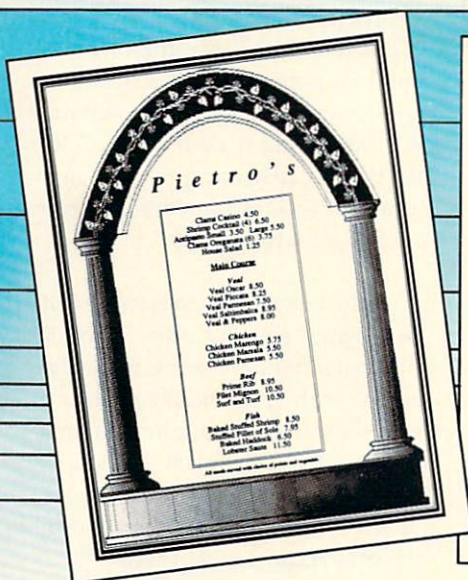
The program should also include some method for unattended printing of a series of graphs. As it stands, it takes a considerable amount of time to print out graphs for even a small number of equities. Batched printing would allow you to print your graphs

while you went to work, and then have them ready when you came home.

Investor's Advantage may not operate smoothly if you are very low on memory. Of course, this shouldn't be a serious problem, since anyone

(continued on page 34)

NEW RELEASE
Affordable Desktop Publishing

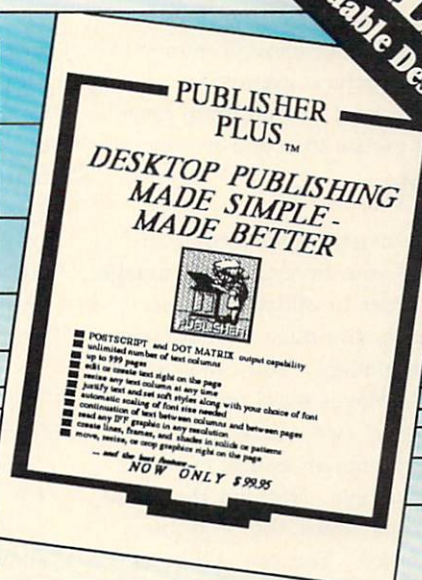


At Northeast Software
Group, Minor Miracles
Are Easy,
Major Miracles
Take a Little Longer.



It took a little while longer for us to get
PUBLISHER PLUS to the point where you
could make some pretty amazing pages — like
this one. But, it's pretty clear that it was well
worth the wait. That's why PUBLISHER
PLUS is quickly becoming the most popular
desktop publishing software for the Amiga
500, 1000 and 2000.

PUBLISHER PLUS



Publisher Plus outputs to a wide range of printers: any preferences-selected printer, dot matrix or any PostScript laser printer or typesetter.

See What You Can Do With

We listened. To Publisher 1000, we added more capability; increased the speed, eliminated the copy protection, reduced the price, and kept the easy-to-use interface. We now call it **Publisher Plus**.

It's a perfect fit for the Amiga 500, although it also fits well with the Amiga 1000 and 2000. Quickly you can create all kinds of printed material — newsletters, signs, reports, menus — without a complication or expense.

With **Publisher Plus**, text is easy to type. Just draw a box, any size or shape, like a newsletter column. Either type directly into the box, or read the text from a word processing program. It's that easy.

PUBLISHER *plus*

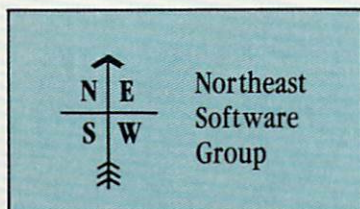
\$99⁹⁵
introductory price

To add graphics, simply draw with Publisher Plus' patterns, or create your own with the pattern editor. IFF pictures and scanned images can be resized or cropped to fit your layout.

Get a close-up view using "smooth-scrolling" or take a full-page view. When you're ready, print it. **Publisher Plus** includes an entire set of fonts for dot-matrix and PostScript printers. If you don't have a laser printer, save your PostScript layout on disk and then print with someone else's laser!

It's for the novice. It's for the expert. **Publisher Plus** — it's for you. Try it today.

Developed by:



Publisher 1000—PLUS...

- ▶ PostScript laser support
- ▶ No dongle
- ▶ Automatic font sizing
- ▶ Cut/copy/paste text
- ▶ More patterns
- ▶ Improved performance

See your local dealer or call:

**Brown-Wagh
Publishing**

1-800-451-0900

1-408-395-3838 (in California)

16795 Lark Ave., Suite 210, Los Gatos, CA 95030

considering investing in the stock market should have enough money to go out and purchase expansion memory. At least this program gives you a new excuse to spend money on your Amiga.

Perhaps the most serious limitation I encountered with Investor's Advantage is that, in order to utilize the Auto Update mode, you must have a Hayes Compatible modem. You can't get by on a pseudo-Hayes like I own. Or, more properly, *used* to own—this program gave me an excuse to buy a Hayes Compatible. Without the Auto Updating capabilities, the program is severely limited. You can still enter all records by hand, but this can be quite time-consuming.

Before you purchase this program, you should either determine the extent of your modem's Hayes compatibility, or (preferably) try out the program at your dealer's, using your modem. (At press time, I was informed by David Miller, author of Investor's Advantage, that versions after 1.05 would also support the Avatex modem, one of the most popular modems purchased for the Amiga. I was not able to confirm Avatex compatibility for this review.)

One final point to consider before you make your purchase decision is the added cost of information retrieval. If you are a serious investor, you should not be swayed, as the annual cost for weekly quotes on thirty equities costs about \$40.00. This cost, when combined with a \$48.00 subscription fee, is comparable to the commission on a single stock purchase.

Things Get Better

Although the program has a number of minor flaws and restrictions, I feel that these are outweighed by the advantages. The best declaration I can make is that the program works. A business program like this doesn't

really need to be flashy. Investor's Advantage is very easy to use and provides rapid information assimilation. It isn't as timely as a tic-chart, but for most investors it will prove more useful. It allows the investor to ignore short-term price fluctuations (which they usually can't do anything about anyway) and focus on longer-term trends. This is probably the safest method for the private investor to use.

The program is extremely user-friendly, allowing most operations to be performed using only the mouse. The Auto Updating utility may be the most valuable feature of the program, allowing unaided updating of your information. I did notice that the Auto Updating became somewhat confused when dealing with my

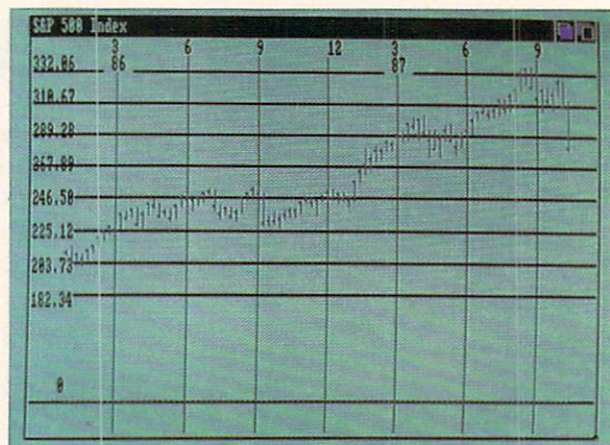


Figure Three: S&P 500 Index

"noisy" phone line. The program often received garbage signals and then had to log off. I had to use the delete history function to delete the bad records, and then log on again. This was not a major problem, but you should expect a few difficulties if your phone system often has a lot of static.

The graphs created by the program are not of the highest quality imaginable, but they are certainly adequate. For the serious technical investor, this program is indispensable, providing the same information as most newsletters do, but on a more timely basis.

Summary

As much as I like Investor's Advantage, I hesitate to give it a blanket endorsement. The program certainly does everything it claims to do. It is also priced fairly and provides an elegant method for updating files and providing useful graphs.

My main reservation concerns what the program actually does. Blind faith in a system you are not completely familiar with is financial suicide. Tracking technical indicators is only one available investment method, and is not necessarily the correct one. I did not feel that it was fair to address this topic in the actual Investor's Advantage review, so I have given a brief overview of different investment strategies in "A Poor Man's Guide to the Stock Market."

If you feel comfortable making investment decisions based on technical indicators, or even combining this information with other criteria to help you make decisions, then by all means purchase this program. If Investor's Advantage does nothing more than help you make the correct investment decision once a year, then it has paid for itself.

About the Author

Richard Knepper is a registered Commodities Broker. He has an M.B.A. in Finance from the University of Wisconsin. He works as a consultant for J.L. McKinzie & Company, a commodities brokerage firm in Chicago. The opinions expressed in these articles are not investment advice.

•AC•

Investor's Advantage \$99.95

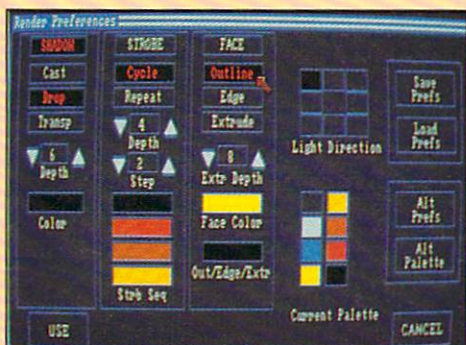
Software Advantage Consulting Corp.
37346 Charter Oaks Blvd.
Mt. Clemens, MI 48043
(313) 463-4995

TV*TEXT

Character Generator



- ▶ shadows, 3-D, strobe
- ▶ adjust light depth and location
- ▶ italics, bold and underline



- ▶ easy-to-use menus
- ▶ select shadows, colors, lighting
- ▶ choose palettes and colors

SAVE \$50,000! TV*TEXT brings capabilities of the most expensive character-generators to you and your Commodore-Amiga personal computer. Pocket all that money while you create professional quality lettering for presentation graphics, or live video production with Genlock. TV*TEXT supports the mouse, medium/high/overscan screen resolutions, the full Amiga palette of 4096 colors, all IFF images, and NTSC/PAL video standards.

You can use any Amiga fonts, such as Zuma Fonts, workbench fonts, etc. Spacing can be adjusted and characters can be stretched, squeezed or even rotated! Text can be positioned with left/right justification or centering.

Make titles exciting with rendering attributes such as italics, bold, underline, outline, edge, extrude (3D), cast/drop shadows and strobes. Create attractive backgrounds using wallpaper or tile patterns. Then captivate your audience with special effects made by applying those attributes to lines, boxes, circles and ellipses.

If you want to make your picture look special, try TV*TEXT!



- ▶ wallpaper or tiled backgrounds
- ▶ stretch, squeeze, rotate text
- ▶ brushes get rendered too



- ▶ extruded (3-D) with drop shadow
- ▶ horizontal lines with outline & shadow
- ▶ different font styles and sizes

Developed by

Zuma Group

Incredibly priced at:
\$99⁹⁵

Other products by **Zuma Group**:
Zuma Fonts Vol. 1, 2, 3 - \$34.95 each
TV*SHOW - \$99.95

See your local dealer or call:
Brown-Wagh Publishing
1-800-451-0900
1-408-395-3838 (in California)
16795 Lark Ave., Suite 210, Los Gatos, CA 95030

The Ultimate Video Accessory *Part II*

Adding computer graphics to your videos is easy, but you must take careful steps to get the highest quality.

by Larry White

You've seen videos ranging in caliber from home movies to your favorite TV shows; from professional business presentations to music videos and slick, lavishly produced commercials. What makes some video productions more polished and exciting than others?

Computer graphics are part of the answer—from the simplest title or "800 number" floating innocently over the action, to fully animated 3D graphics that can take you to places existing only in the imagination of the artist. When properly used—not overused—computer-generated video effects can add a lot to a video production.

Video production houses may spend tens of thousands of dollars for computer graphics systems, but for only a few thousand dollars, you can generate many of these sophisticated effects with a Commodore Amiga and some sophisticated software. Programs for Amiga video applications range in price from under \$50, to as much as \$299.

Let's take a closer look at some of these applications, and the basic software, computer hardware, and video equipment you'll need.

Titling and Character Generation

You've just returned from an exotic vacation, and you've shot a few hours of exciting video with your new Camcorder. When you show the tapes to your friends, they're polite, but you sense you never really grabbed their full

attention. Grabbing the audience with your opening shot is a good video technique. Let's boot your Amiga and add some touches that will make your friends ask to see your video again and again.

With a little imagination, you can use almost any program to create a title screen. Let's begin with the simplest title screen, a computer generated screen. You'll record the screen exactly as it appears on the computer monitor and insert in front of your original video (camera or Camcorder image). Normally, this piece is inserted as you copy and edit your video onto a second video cassette recorder. You should leave a few extra seconds of blank tape at the beginning, which you can later replace with the title screen. Most professional videographers do not record on the first 30 to 60 seconds of a tape, allowing for proper tape threading in the VCR and insuring that the machine is up to proper speed for copying and editing.

Title screens should be simple, with large text for easy reading because they appear for only a few seconds (See Figure 1.). Any paint program can be used to create a title screen, although I prefer one with good text handling, such as Deluxe Paint II. I like to add larger fonts (such as the Zuma fonts) for clean, easy-to-read titles. Express Paint promises to fill this criteria with color fonts available in their next version; I'll comment more after I try it first hand. Remember, your final quality will probably be far below what you're accustomed to seeing on an RGB Amiga monitor.

Many programs are specifically designed for video titling applications. JDK Images CG1 (Character Generator 1), emphasizes crisp, clear type in bright colors with appropriate shadowing for emphasis (See Figure 2.). Up to 100 screens of information can be stored and cycled using various transitions for effect.

Aegis Titler, TV Text, and Deluxe

Titler (formerly Station Manager EF/X) are all designed specifically for this type of operation. Some programs, like Deluxe Titler, offer animation capabilities which let you fly text or graphics across the screen. (You can get similar results with Deluxe Video, but you can do so much more with this program that it will get special attention in another article.) Like most software, each program has strengths and weaknesses, as well as various price points. I'll go into each program in detail later in this series.

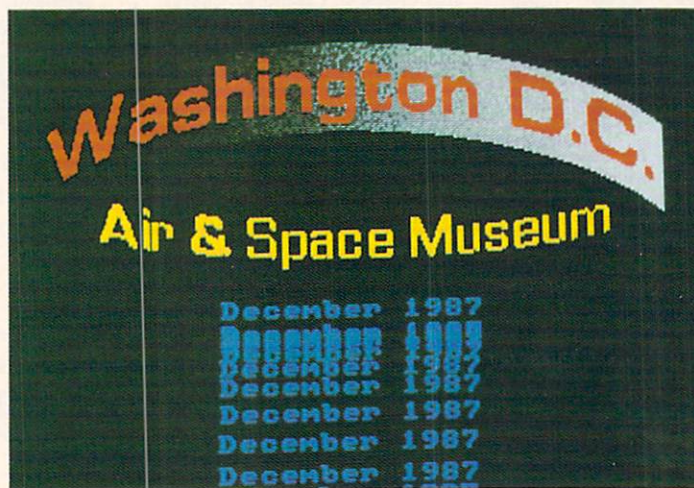


Figure 1: Simple title screen created using DeluxePaint II. Text uses Zuma fonts for large, clear titles.

(continued on page 38)

**SPECIAL EFFECTS
SLIDE SHOW**

TV SHOW

Now you can produce your own animated presentations on the Commodore-Amiga or video tape. Use any IFF pictures and over 50 exciting transitions including rolls, reveals, wipes, flies, fades, color cycling and more. Each picture can remain on screen for a preset time or until a keyboard entry is made. Now really get fancy. Fly or wipe object brushes onto the screen. Experiment all you want — you can play any portion of your script at anytime during your editing session! Incredibly easy. Lights, camera, ACTION!

**SHOWCASE YOUR
COMPUTER PICTURES WITH
TV*SHOW**

FULL COMPATIBILITY FOR FULL FLEXIBILITY

- TV*SHOW is compatible with Amiga 500, 1000, and 2000 models (1 MB recommended)
- All IFF graphic formats including HAM
- Overscan screen sizes of over 700 x 480 pixels
- NTSC/PAL video standards
- Genlock for video production

Developed by



Zuma Group

Incredibly priced at:

\$99⁹⁵

Other products by **Zuma Group**:
Zuma Fonts Vol. 1, 2, 3 - \$34.95 each
TV*TEXT - \$99.95

See your local dealer or call:
**Brown-Wagh
Publishing**

1-800-451-0900

1-408-395-3838 (in California)
16795 Lark Ave., Suite 210, Los Gatos, CA 95030



Figure 2: CGI by JDK Images produces clear, crisp titles with shadows, lines and patterns available. Keep title clean and simple to be effective.

After you've created and stored your title screen on disk, how do you transfer it to video tape, and what kind of video quality can you expect?

Technical Considerations

The amount of detail in an image is determined by the resolution. In video, the vertical and horizontal resolution are seldom equal, since they are controlled by different factors. Vertical resolution is limited by the 525 scan lines of the NTSC video system. Since all these lines do not appear on the screen (They can carry test signals, closed caption information, even communications between TV stations.), the actual limit for vertical resolution is below 500 lines. Horizontal resolution depends more on the amount of information carried in the video signal (bandwidth), and other factors, like the interaction of colors and the precision of electron beams and phosphors.

The total amount of information which can be produced is calculated by multiplying the horizontal resolution by the vertical resolution. The result is the total number of picture elements (pixels) in the system. The more pixels you have, the more detail in the overall scene.

The Amiga controls its own resolution, at least when it leaves the computer. Amiga resolutions range from lo-res (320 horizontal X 200 vertical), to hi-res (640 h X 400 v). Not all software can work with all resolutions. The more resolution and colors in the image, the more Amiga memory you need to manipulate those images.

The Amiga offers a choice of resolutions, but your video recorder has a specific limit. Most consumer-grade VCRs can typically record between 200-230 horizontal lines, although the new super VHS and ED (extended definition) Beta recorders can record

Unfortunately, resolution is not absolute. If you copy an image with 640 horizontal lines onto a VCR with a maximum horizontal resolution of 250 lines, the recorded result contains less than 250 lines. This discrepancy is caused by signal loss in the cabling and other image factors. It's important to know the best you can hope for (in this example) is only 250 lines.

Another problem is that video information is stored on tape as an analog signal, unlike computer data which is stored digitally. Each copy is subject to quality loss, limiting the number of generations (copies of copies) which you can make with reasonable quality results.

You may have seen ads for digital VCRs; although they may deliver slightly improved picture quality, they are just analog recording systems which use digital processing to provide a better picture and digitally processed special effects.

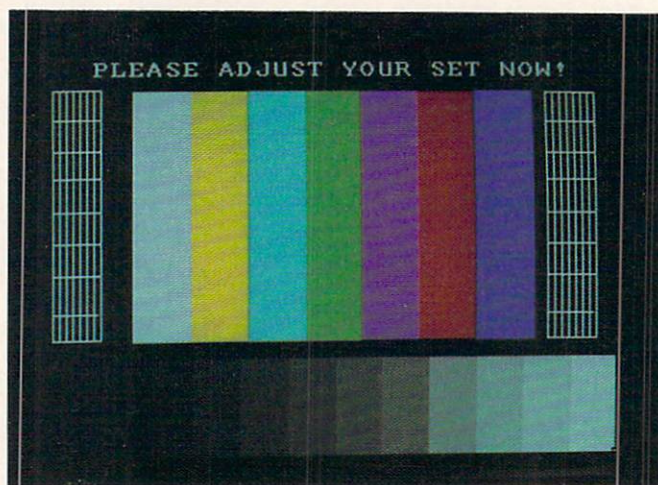


Figure 2a. Colors bars, placed on the tape before the first scene or title, let you adjust the TV or monitor for optimum performance. This one was made with Deluxe Paint II.

details as fine as 400 or 450 lines. This isn't as bad as it sounds, when you consider the best current broadcast signals in the US don't exceed 330 lines of horizontal resolution.

Of course, there's more to video image quality than resolution and pixel count. Color rendition and interimage effects can also affect the final picture. Similar to resolution, the more generations you copy, the worse these effects.

Don't be disillusioned by the lack of quality in your video image. You must understand

these factors, so you can control them and produce the highest possible quality your equipment allows. You can take easy and relatively inexpensive steps to improve the video image: You can use shielded cabling with gold plated contacts wherever possible,

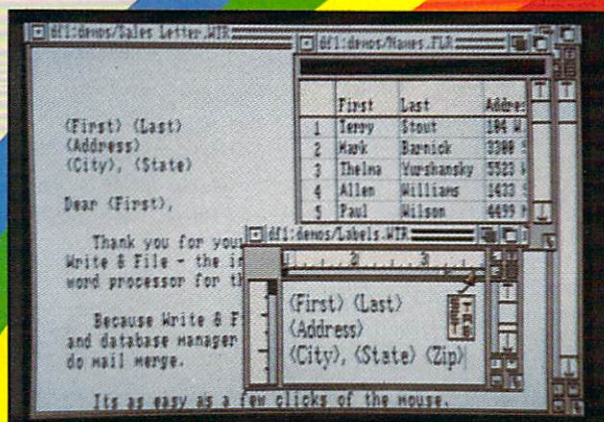
(continued on page 40)

Softwood WRITE & FILE

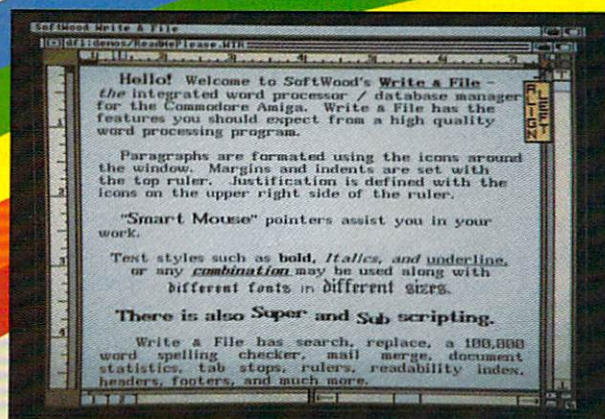
Integrated Word Processor/Database Manager

Multiple Font Styles, Spell Checker and Powerful Features make **Write & File** a SUPER Word Processor. The Integrated Database Manager for Reports and Mail Merge Make it SUPERB!

- Features unique "Smart Mouse" and "Undo"
- Search/replace; headers, footers; hor/vert. rulers
- Prints in graphics (pretty) or text (fast) modes
- Includes over 100,000 word dictionary
- Supports foreign letters, paper size, currencies



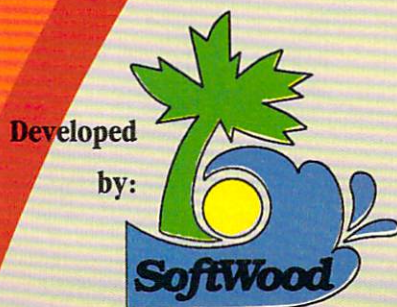
Manage lists while writing document



WYSIWYG—uses any Amiga font styles and sizes

- Views multiple database records at a glance
- Manages data in rows/columns like a spreadsheet
- Sorts and searches on multiple criteria
- Calculates fields using math formulas
- Merges lists with documents for labels/envelopes

The First to Provide a DOCUMENT READABILITY INDEX!



Developed
by:

Incredibly priced at:

\$99⁹⁵

See your local dealer or call:
**Brown-Wagh
Publishing**

1-800-451-0900

1-408-395-3838 (in California)

16795 Lark Ave., Suite 210, Los Gatos, CA 95030

and use the shortest possible cables throughout your system. If you want to go a step further, an under-\$70 signal booster (often misnamed "enhancer") can cut down on signal losses when making copies. There are also more expensive ways of processing the image which will be discussed in another article.

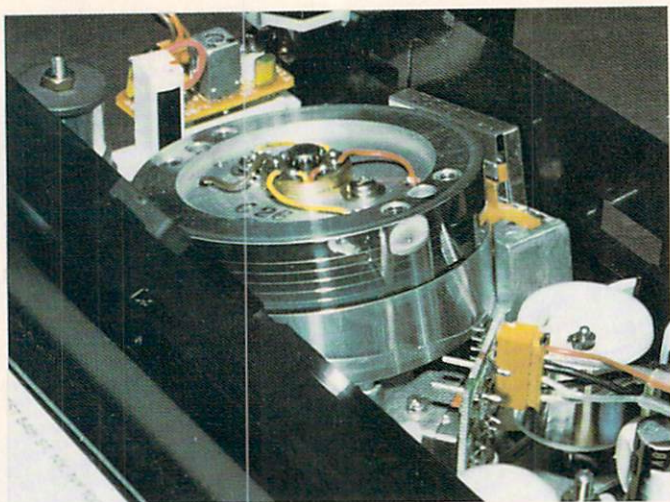


Figure 3. Video heads are located inside a large, silver drum. Tape is wrapped around and moves by as the drum rotates.

VCRs for Desktop Video

A more immediate concern regarding our desktop video applications is the quality of the video image when you stop and start the VCR. Many VCRs produce a short burst of video noise when you start a new recording. This would cause an unacceptable result each time you cut (switch) to a computer-generated graphic. To understand why this happens, we need to discuss how VCRs record images.

An audio recorder works in linear fashion. The tape moves across the magnetic recording head in a straight line. The faster the tape moves, the more information you can record because more magnetic sensitive media (audio tape) passes by the head at any given time. Unfortunately, video information is much more complex than audio, and video must also contain a sound track to be effective. To capture all this information on a moving tape, the tape must either be extremely long as compared to the same play time for audio, or very wide (Professional video is often recorded on one inch tape.).

Wrapping the tape around a drum at a slight angle and spinning the heads as the tape travels past the drum is

another solution. By placing several heads on the drum, you can pack information tightly on the tape diagonally, known as helical recording. The audio track is still recorded in a linear fashion using separate heads placed along the edge of the tape path. (See Figure 4.) This can be used to great advantage, since many VCRs feature audio dubbing, which lets you replace the sound track independent from the video. Some VCRs advertise stereo sound; a second sound track is recorded helically between the video tracks, but when you audio dub, only the linear sound track is replaced.

Here's the catch: To erase any existing information on a tape before recording new information, the tape passes by a special head called an "erase head." In most consumer VCRs, this head is stationary and can't erase

with enough precision along the helical path. The solution involves using a "flying erase head" which spins along with the normal recording heads. Although the technology is rather simple and straightforward, it can add unnecessary expense for a feature not needed for most applications (which is why these heads are often not included in today's highly competitive models). See Figure 5 for a comparison of normal and flying erase heads. Most 8mm recorders, and many top of the line models in all formats, offer flying erase heads. If you're buying a VCR for desktop video, I'd strongly recommend you insist on this feature.

If you already have video equipment, or can't afford to go after the more expensive models, you can still create clean edits. Whenever possible, keep the recording VCR in record mode, using the pause control (instead of "stop") to freeze the tape between scenes. Use a video switcher to cut from one video source to another without stopping the record operation. Some of these units even offer signal processing and special effects (fades, wipes, and dissolves).

Now that you have a better understanding of how the VCR works, let's discuss how to get the video image

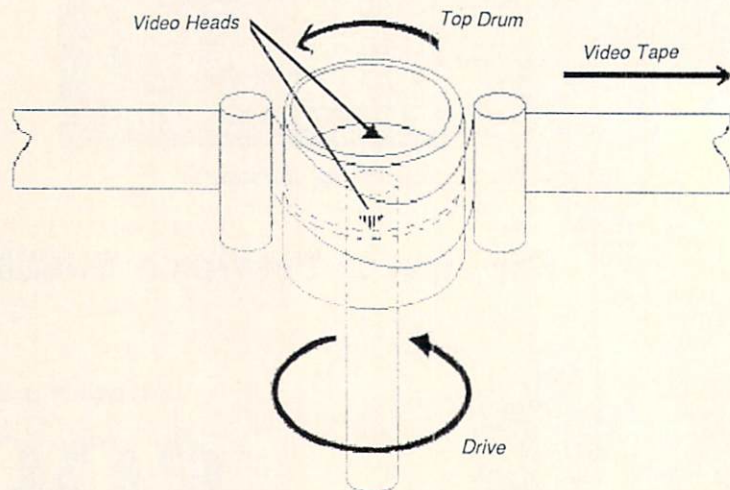


Figure 3a. Recording heads are shown relative to drum and tape. The bottom drum is stationary with a sloped guide to help the tape cross the drum at an angle.

from your Amiga to the VCR. The specific hardware is somewhat dependant on which Amiga you use, but the principles involved are basically the same. A1000 users need a composite NTSC

video output jack as part of their basic equipment. Much has been written about the overall quality of this output jack; it's good enough for most non-critical purposes. It is electrically 100% compatible with the video-in jack on your VCR, although sometimes you might need to change the connectors on the end of the cables. The stereo audio outputs are also compatible, but if you're not using a stereo VCR, you'll have to tie them together the same way you would use your Amiga 1080 monitor.

The A500 and A2000 also have NTSC video output jacks; however, on these models, the output is black and white

only. On all three Amiga models, you'll be better off if the video signal is taken from the RGB port; although to be recorded, the signal must first be converted to NTSC composite video.

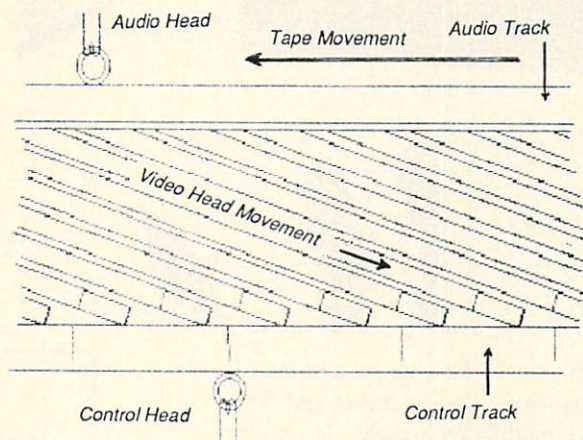


Figure 4. Video information is stored along diagonal tracks, while the audio track is linear along the top. The VCR uses a control track to adjust playback speed to match recording speed precisely.

Several devices fit this purpose. I've been using a prototype of Commodore's A520 Modulator which converts RGB to composite video, or mixes the image with the audio output to produce an RF (radio frequency)

signal. You need this signal if you're using the Amiga with a television set or an unusual VCR that doesn't have a composite video input. Rating the signal quality from best to worst, they are RGB, NTSC composite video, then RF.

The Commodore modulator is designed for people who want to use an Amiga with a monitor that doesn't have analog RGB inputs, and accordingly, they don't have an RGB passthrough. To monitor what you're doing on the computer, you must use a standard video monitor. If you're using the Amiga 1080, you can switch it to composite video, but you must connect the video signal to the composite video input on the monitor's back. Remember, the image won't be nearly as sharp as you've learned to expect. If you're planning to use any complex graphics, design and store them using the RGB interface, turn off the computer, connect the video interface, and reboot.

Many A1000 users are already using the Commodore Genlock device for video output. With genlock, you can synchronize your Amiga graphics to an incoming video signal and have

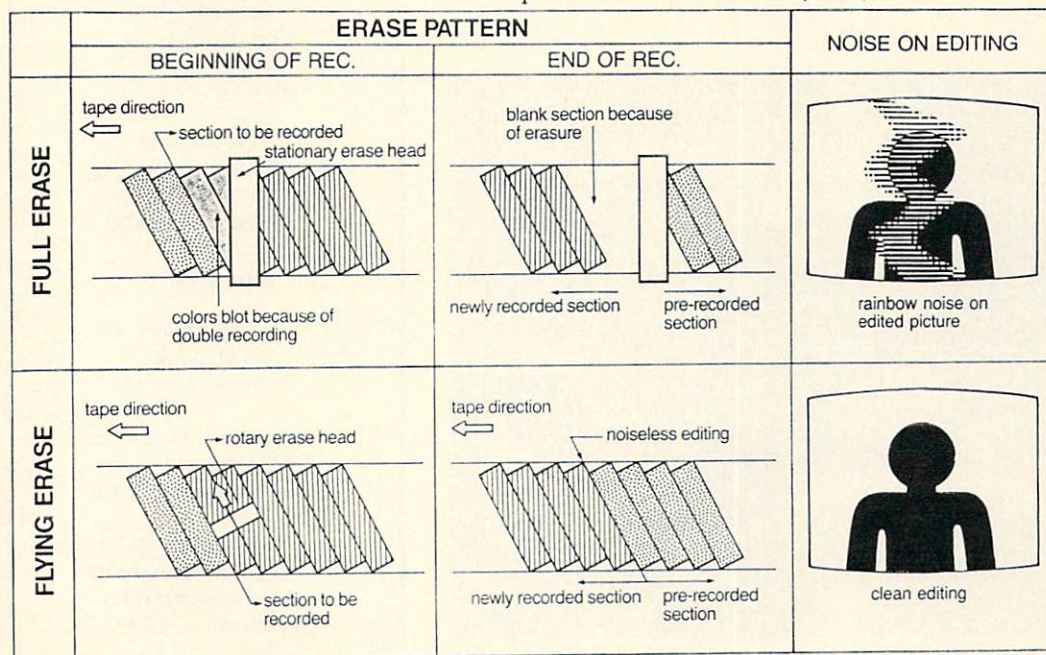


Figure 5. This comparison of video erasing methods shows the noiseless result of flying erase heads. (Courtesy Sony Corp.)

them both on the screen and in your video recording at the same time. Genlock is one of the most exciting techniques in desktop video, and I'll go into much more detail in a future installment. Even without using the genlock feature, several of these devices offer composite video out and RGB passthrough to your Amiga monitor. Mimetic's AmiGen is an under-\$200 genlock device with RGB and NTSC composite video output that works with all Amigas. I received AmiGen just as I was closing out this article and look forward to trying it out.

(continued)

If you haven't purchased your VCR, there are a few more technical points to consider. A few VCRs have a jog mode which lets you move back and forth one frame at a time. These VCRs are usually professional or top-of-the-line models and tend to be expensive side. Frame advance is usually controlled by a large knob on the front of the VCR. This is a handy feature for editing video.



Figure 6. Ambico's Super Duper reduces signal loss when making video copies. Notice the thick, shielded cables with gold plated contacts. The separate power supply is not shown.

If you're planning to do a lot of video editing, you'll want several VCRs which can be connected to an edit controller, which allows you to work several VCRs from a single set of controls. The Amiga itself may become your edit controller in the near future, as the hardware and software become available.

If you want to push your Amiga's capabilities to the maximum, you'll want a VCR which can record a single video frame. Creating a number of frames, one at a time, and recording them individually on the VCR lets you stretch the Amiga's capabilities even further. You can create complex computer graphics using all the tricks, regardless of how long it takes the Amiga to draw the final image. Some complex images using ray trace techniques, especially with anti-aliasing subroutines, can take several hours to complete.

Very few VCRs have still frame capability, but both Panasonic and Sony have models with both still frame recording and jog mode. The Panasonic AG1950 (VHS) and Sony SL-HF1000 (Super-Beta Hi-fi) each sell for about \$1000.

Even with far less sophisticated VCRs, the Amiga can do a lot to enhance your videos. Next time we'll take a closer look at some of the available software as we begin to make a video.

•AC•

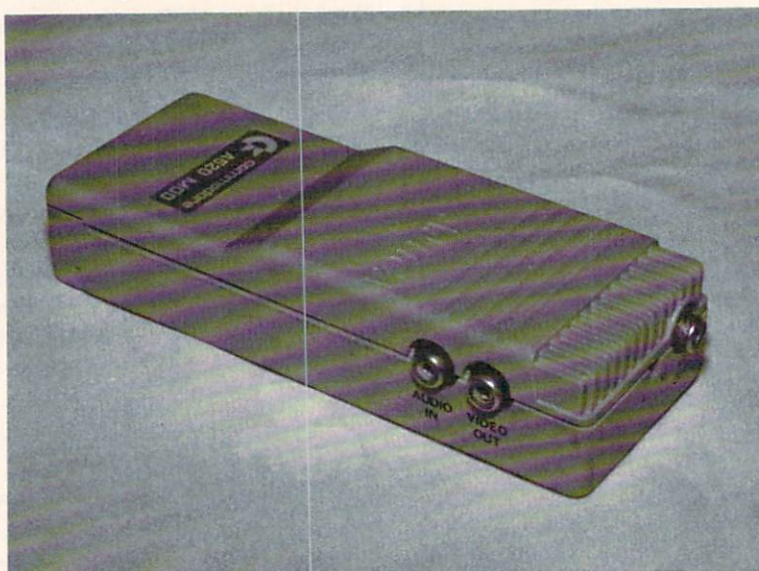


Figure 7. Commodore's A520 modulator lets you get NTSC composite video or RF signal from A500 or A2000. A small box plugs into the back of the computer, instead of the RGB cable.

Aegis Titler
Aegis Development
2210 Wilshire Blvd. Suite 277
Santa Monica, CA 90403

CG1 By JDK Images
PVS Publishing
3800 Botticelli Suite 40
Lake Oswego, OR 97035

Deluxe Paint II
Deluxe Titler Deluxe Video
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404

Express Paint
PAR Software Inc.
P.O. Box 1809
Vancouver, WA 98666

TV Text Zuma Fonts
Brown-Wagh Publishing
100 Verona Court
Los Gatos, CA 95030

Hardware:

Ambico Inc.
50 Maple St.
Norwood, NJ 07648

Commodore-Amiga Inc.
1200 Wilson Drive
West Chester, PA 19380

Panasonic Company Matsushita
Electric Corp of America
One Panasonic Way
Secaucus, NJ 07094

Sony Corp of America
Sony Drive
Park Ridge, NJ 07656

PACKED WITH POWER!

Three of the BEST utilities for your *AMIGA*® !



\$79.95

ZING!® is a collection of *AMIGA*® utilities which combine the powerful CLI commands with the friendly Workbench environment. Files can be displayed and manipulated (e.g. copying, moving, deleting) with the mouse. All of the basic system commands (available in CLI) have been carefully redesigned into mouse, menu, and function key operations. You can selectively copy files and directories from entire disks in a single step! In addition to enhancing and simplifying the normal capabilities, **ZING!** provides an integrated collection of new tools for the *AMIGA*. Included are a task monitor, print spooler, screen dump tools, and much more! Everyone, from the beginner to expert, will find using the *AMIGA* easier and more productive with **ZING!**

ZING!Keys® is a sophisticated reprogrammable MACRO and Hot Key program. A program which can be used with any programs in the multitasking environment. You can train **ZING!Keys** to accomplish the most annoyingly repetitive tasks in a much easier fashion. You can program any key stroke to type out any series of commands or text. You can even record mouse movements and play them back as a single key stroke! All MACROs and Hot Keys can be used from within any multi-tasking program on the *AMIGA*. Save time NOW by ordering **ZING!Keys**!



\$49.95

THE DEMONSTRATOR

How many times have you demonstrated your *AMIGA* to your friends or potential buyers? **The Demonstrator** is the answer to your dreams! This new product will record your demonstrations and allow you to play them back automatically! You can add text windows, subsections, and speech to produce sophisticated tutorials. You can control the speed or STOP the playback, or lock out the keyboard so no one can interfere with the demonstration. You can cause the demonstration to repeat itself automatically! Buy **The Demonstrator** today!

Only \$39.95!

**MERIDIAN
SOFTWARE
INC.**

P.O. Box 890408
Houston, TX. 77289-0408

(713) 488-2144

Credit Cards and Dealer
Inquiries Welcome!

AMIGA is a registered trademark of Commodore-AMIGA, Inc.
ZING! is a registered trademark of Meridian Software, Inc.



Amiga is a trademark of Commodore-Amiga, Inc.

AMIGA Dealers!

We are a Dallas Distributor exporting to customers in over 20 Nations around the globe.
Call us today at 1-800-531-4747 (Overseas call 214-6601763) and ask for your Dealer Kit!
Here is just an overview of some of the AMIGA products we stock.

PRACTICAL PERIPHERALS

Practical Modem 2400 SA

Hayes compatible 2400 Baud Modem

BYTEC

Smart products for a smart Computer!

The first intelligent Diskdrive for the Amiga!

This drive serves as DF1: or DF2:

A special intelligent mode allows to disable the drive and regaining some memory without disconnecting the drive!

Compatible with A500 / A1000 and A2000.

Users: Call for Info and Dealer next to you!

Flicker Master Flicker Master Flicker Master

Reduces Interlace Flicker dramatically.
For the price-conscious customer who doesn't want to buy a high persistence monitor.

Sargon III

Huge Print

If you like it BIG this is it;
print Up to 8.5 x 13 feet . . . !

LASER SOUND

PCM-Sound samples for the AMIGA!
IFF format; compatible with most
Sound editors for the Amiga!



InterComputing Inc.

2100 N. Hwy 360, Suite 2101
Grand Prairie, TX 75050-1015

Order line: 800-531-4747
Technical Support: 214-988-3500
International calls: 214-660-1763
Telex #: 4932133 ICINCUI
FAX: 214-660 3695

We stock a full line of Amiga soft and hardware since 1985, when the machine was first introduced, and serve customers in more than 20 Nations around the world. Most of our employees own an Amiga and use the available products.

We may be new to you, but we're experienced with the Amiga!
Check our prices, and you'll see that we try to be quite fair. Good products are worth the money, and you will not be disappointed with our service.

We are a member of the Dallas Better Business Bureau.

City Desk

Lots of desktop publishing
at a small price!

Synthia

Excellent sound synthesizer!



Butcher

Very powerful conversion & processing utility for digitized graphics!

Starboard2 Memory cards:



Versions for the A500
A1000
and A2000

Sculpt 3D and ANIMATE 3D

and more (please call for a complete catalog)

BYTEC

Smart products for a smart Computer!

MIDI-X

Great looking Midi Interface that fits all Amigas. Simply use different cables to connect the Unit to the CPU.

Because it's less expensive to have one MIDI with two cables than 2 different MIDI interfaces!

By Gerald Hull

"... From all we have learnt about the structure of living matter, we must be prepared to find it working in a manner that cannot be reduced to the ordinary laws of physics."

—Erwin Schroedinger, *What Is Life?*

In 1970 mathematician John Horton Conway created "The Game of Life," a mathematical recreation based upon cellular automata theory. Life generates a special universe of objects exhibiting complex and seemingly unpredictable behaviors. In addition, Life illustrates principles being applied today to fields as diverse as cosmology, particle physics, and thermodynamics.

Part I of this article discussed the origins of Life in cellular automata theory, its rules, its history on the Amiga, and some of its curious properties and implications. In Part II, we continue by taking a detailed look at the efficient use of the Amiga blitter.

The Valley of the Docs

As the world is gradually discovering, the Amiga is a remarkably sophisticated machine. End users have a harvest of software to choose from, but the programmer confronts a formidable learning curve. There is helpful documentation available for programmers, of course (most notably provided by Commodore).

To learn how to use the blitter, you will find a great deal of useful information in the *AMIGA Hardware Manual*. If you have the Addison-Wesley update—*AMIGA Hardware Reference Manual*—all the better. It was extensively rewritten and updated, and contains some illuminating diagrams. At the same time, it contains some very misleading language, which has conspired to give the blitter a kind of "occult" status.

For example, at one point the manual states, "Although the hardware deals in words for pointers and modulus, the values loaded into these hardware registers from the 68000 are treated as byte counts." (p. 170) "What else loads values

into those registers?" I wondered. Here's what the sentence really means: The addresses and offsets you provide must be even numbers.

Similarly, the discussion of the bitplane addresses strongly suggests that they must be split into two halves: "The pointer for source channel A has two register addresses. BLTAPTL contains the low-order part (bits 15–0) and BLTAPTH contains the high-order part (bits 18–16) of the pointer address." (p. 168)

As a result, even some of the most illuminating public domain source involving the blitter laboriously splits bitplane addresses into two halves. That's completely unnecessary. As the LIFER code makes clear, the bitplane addresses fed to the blitter registers can be treated as 32-bit longwords, just like any other 68000 addresses.

Here, as anywhere else, I've found the best approach is first to read all the documentation and examples you can find. Then add a judicious amount of common sense. This usually restricts the possibilities sufficiently enough so experimentation ("Dancing with Mr. G") can yield a solution. Anyway, that's how I got LIFER to work.

Banging The Registers

Presumably, the only absolute address in AmigaDOS is ABSECEBASE at 0x00000004. Actually, all the blitter registers are also available as offsets to a specific address = CUSTOM = 0x00dff000. You do not actually use that number, of course (ha ha). AMIGA.LIB supplies the right value to the linker for an assembler XREF to "_custom" or C reference to "custom."

The CUSTOM.H and CUSTOM.I files list all the blitter registers. When included in your program, they automatically provide the proper offset values. People who should know say there are registers here that nobody has ever used. The LIFER program uses only a few of them. The *AMIGA Hardware Manual* discusses a number of additional registers and capabilities.

(continued)

The blitter registers used by LIFER are listed in Table 1. BLTCON0 and BLTCON1 contain a variety of special flags the program packs together into one long word, CON. ASHIFT and BSHIFT will be discussed at length later. USEA, etc., tell the blitter whether or not to regard the value of BLTAPT, etc., as a bitplane pointer. They're obviously needed because zero is a valid address.

The MINTERM is the logical function of source bitplanes A, B, and C which you want put in destination plane D. Make sure that if USEC is 0, for instance, your minterm does not distinguish between C and not-C. In other words, the minterm, 0x80 in this case, would be defective because, when A and B are both 1, the value of D would be contingent on the value of C.

(bitplane use)	1	1	1	0
	D	A	B	C
	1	1	1	1
(should be 1)	0	1	1	0
	0	1	0	1
	0	1	0	0
	0	0	1	1
	0	0	1	0
	0	0	0	1
	0	0	0	0

BLTxPT (x = A,B,C,D) are pointers to the bitplanes involved in the blit. As noted, they should be even values; like the 68000 stack pointer (SP = A7), they do not admit odd counts. If the addresses you poke in them result from standard AmigaDOS memory allocation routines (such as AllocMem and AllocRaster), you are guaranteed even values.

BLTxMOD (x = A,B,C,D) represent "modulos," necessary for dealing with subportions of images. They specify how many bytes of memory the blitter should skip over before it grabs the next sequence of pixel values. Since the blitter deals with words only, these modulos must be even, too.

Finally, the BLTSIZE word tells the blitter how much memory to move. The first 10 bits provide the number of rows to move, each to consist of the number of words given in the last six bits. If you have specified a non-zero modulo, the blitter then skips that specific number of bytes before grabbing the next row. BLTSIZE is quite special. The movement of a value into this register is what actually triggers the blit.

Dancing in the Web

The male black widow spider, looking for romance, must be very careful when he enters the female's web. To avoid being mistaken for yummy prey, he approaches by stepping in a peculiar rhythm. You must approach the blitter on the

Amiga in the same way. The multitasking operating system constantly uses the blitter for other purposes. If you make a false step, Ms. Black "Guru" Spider eats your program.

Consequently, AmigaDOS provides procedures that can ensure your blits are "harmless." The basic sequence is represented by the following C code fragment:

```
...
...
OwnBlitter();
for (i = 0; i < NUMBER_OF_BLITS; i++) {
    WaitBlit();
    FUNCTION_THAT_INVOKES_THE_BLITTER(&BLIT_NODE);
}
WaitBlit();
DisownBlitter();
...
```

The functions—OwnBlitter(), WaitBlit(), and DisownBlitter()—are all supplied by AMIGA.LIB. The first gives your code exclusive rights to the blitter; the second makes sure any preceding blit has finished; and the last relinquishes those rights back to the operating system.

The FUNCTION_THAT_INVOKES_THE_BLITTER consists of a sequence of assignments to the blitter registers which define a particular blit, culminating in an assignment to BLTSIZE:

```
struct Custom *chip = (struct Custom *)0xdff000;

void FUNCTION_THAT_INVOKES_THE_BLITTER(node)
Blit_node_structure *node;

{ chip->bltcon0 = node->bltcon0;

chip->bltcon1 = node->bltcon1;

<all other blitter register assignments>

chip->bltsize = node->bltsize; /* this makes it happen! */ }
```

I've oversimplified somewhat. You must pay more attention to data types and conversion of operandi than this example shows. Nonetheless, once you've hacked out those details, a C routine equivalent to the one above is quite sufficient to make the blitter dance to your rhythm (and not eat your code).

LIFER follows a somewhat more complicated and supposedly more "proper" approach to invoking the blitter. AmigaDOS has built-in machinery for managing and scheduling blitter use. This makes sense for such an essential and limited resource. (This approach is described

starting on page 2-62 of the old *ROM Kernel Manual*, Volume 1. I'm not sure how the Addison-Wesley version handles it.)

Instead of directly accessing the blitter, the program sends the AmigaDOS routine `QBSBlit()` a pointer to a special data structure. In turn, this structure includes a pointer to the actual routine (`_doblit`) that bangs the registers. `QBSBlit()` queues up this request to use the blitter, and calls `_doblit` when its turn comes up.

A Structured Approach

The data structures used in passing information are essential to understanding this reliance on `QBSBlit()`. The AmigaDOS routine expects a pointer to data in the following format:

```
struct bltnode
{
    struct bltnode *n;
    int (*function)();
    char stat;
    short blitsize;
    short beamsync;
    int (*cleanup)();
};
```

The include files `BLIT.H` and `BLIT.I` contain this definition of `BLTNODE`. The only structure elements needed for `LIFER` are `FUNCTION` and `BLITSIZE`; the rest are initialized to zero. `FUNCTION` gets a pointer to the assembler routine `_doblit`, and `BLITSIZE` gets the size of the blit encrypted the same way as for `BLTSIZE`.

How does `_doblit` get the data it needs for banging the registers? By making `BLTNODE` the first element in a larger structure, called `LIFEBLTNODE`, it is possible to "piggy-back" that information so `_doblit` can get to it. `LIFE.ASM` has the assembler version of `LIFEBLTNODE`; `LIFER.C` has a C version. The assembler version contains an extra element (`DUMMY`) to compensate for the fact that C structures are built to longword boundaries.

`QBSBlit()` was designed to allow the programmer to concatenate a sequence of related blits. When control returns to `QBSBlit()` from your `FUNCTION`, it checks the return value, which, in AmigaDOS, means the value in register `D0`. If that value is non-zero, your routine is immediately called again.

My assembler routine `_doblit` takes advantage of this feature and uses the amount in `REPEAT` in `LIFEBLTNODE` to determine how many blits are required to complete the task at hand. Before calling the `FUNCTION`, `QBSBlit()` very conveniently makes sure register `A0` points to the start of

the `CUSTOM` registers (`$00dff000`), and `A1` points to the `BLTNODE` (actually, `LIFEBLTNODE`) structure that it was given.

So, all `_doblit` needs to do is keep banging registers and decrementing the initial `REPEAT` value until the `REPEAT` value reaches zero. Since all the `LIFEBLTNODEs` required for a sequence of blits have been put into an array, `_doblit` simply increments an array `INDEX` pointer to get the data for subsequent blits. An additional variable, `DZIP`, is required because the assignment to `BLTSIZE` scrambles the `D0` register.

The drawback to using the value returned to `QBSBlit()` to perform a whole sequence of blits—`LIFER` needs 14 to complete a Life generation—is the impact this has on multi-tasking. For example, if you attempt to use `CLI` at the same time as `LIFER` is churning away, you find a significant delay between responses to your keystrokes.

I've Got Algorithm

The wonderful people who read and remember Part I of this article may raise their eyebrows at the assertion that `LIFER` needs 14 blits per generation. Am I not the fellow who bragged about doing a Game of Life generation in merely nine blits?

The explanation is five of those blits are needed for setting up the wraparound feature of `LIFER`. If you dispense with this "toroidal" effect, as does Tomas Rokicki in his version of the Game of Life (Fred Fish #31), those five extra blits can be eliminated. I find the wraparound version more aesthetically satisfying. Regardless, the techniques required for providing that feature reveal some interesting aspects of blitter use.

Before going any further, however, let me point out I have taken these techniques (and much else) from a publicly released source fragment of Alonzo Garipey's program `FASTLIFE`. Indeed, most of the code in `LIFER.C` comes either from Garipey or Scott Everndon. As Rosalind Franklin, whose work was crucial to the discovery of DNA, once said, "We all stand on each other's shoulders."

The approach takes advantage of the fact that the state of a cell ("living" or "dead") is mapped onto the screen image which represents it (some 1s or all 0s). Consequently, any logical operations performed on the copy of the screen are equivalent to operations on a two dimensional boolean array representing a borderless Life universe.

The basic idea of the wraparound screen is if you go off one edge, you show up in the same relative place on the opposite side. A lot of computer games, most notably the

(continued)

Table One
Blitter (Custom) Registers Used By LIFER

REGISTER	ADDRESS	SIZE	<PART> / FUNCTION		Remarks
bltcon0	0xdff040	word: bits 15-12 bit 11 bit 10 bit 9 bit 8 bits 7-0	ashift usea useb usec used minterm	amt to shift a-plane (a) should a-plane be used? (b) should b-plane be used? should c-plane be used? should d-plane be used? logical function of a,b,c	(a) Bits in a 68000 word are numbered from least significant bit (LSB = 0) to most significant bit (MSB = 15); when read from left to right, however, the MSB is encountered first.
bltcon1	0xdff042	word: bits 15-12 bits 11-0	bshift <other>	amt to shift b-plane	(b) When set (= 1), the blitter uses the bitplane. When clear (= 0), it won't use it. (See LIFER.C.)
bltatwm	0xdff044	word	a-plane first word mask		(c) These addresses, as with any addresses generated by the blitter in the course of a blit, must lie in the range 0..524288 (0x00000..0x80000), or "chip memory."
bltalwm	0xdff046	word	a-plane last word mask		
bltcpt	0xdff048	longword	c-plane ptr	<even byte address> (c)	
bltbpt	0xdff04c	longword	b-plane ptr	<even byte address>	
bltapt	0xdff050	longword	a-plane ptr	<even byte address>	
bltdpt	0xdff054	longword	d-plane ptr	<even byte address>	(d) Since $2^{10} = 1024$, as does $2^6 * 16$ (bits in a word), the blitter has the inherent capacity to handle 1K x 1K resolution images.
bltsize	0xdff058	word: bits 15-6 bits 5-0	height width	plane height in pixels plane width in words (d)	
bltcm0d	0xdff060	word	c-plane modulo	<even byte amount>	
bltbm0d	0xdff062	word	b-plane modulo	<even byte amount>	
bltam0d	0xdff064	word	a-plane modulo	<even byte amount>	
bltdm0d	0xdff066	word	d-plane modulo	<even byte amount>	

Asteroid™ variants, use such a screen. It's also called a toroidal screen because it is topologically identical to a doughnut or teacup. If this does not make sense to you, don't worry.

```

0000000000
0123456780
0812345670
0781234560
0678123450
0000000000

```

->

```

5678123456
8123456781
7812345678
6781234567
5678123456
8123456781

```

In order to get wraparound results with the blitter, we first prepare a special copy of the bitplane currently displayed on the screen. We blit the 320 x 200 pixel image into the center of a 352 x 210 bit area of memory. As a result, the copy is bordered on right and left by an extra word, and on top and bottom by five extra lines. (Five are needed for the largest "critter" comprising a Life cell.)

Next, we blit the top five lines of the image into the space after the bottom, and the bottom five into the space above the top of the image. We then do the same with the word-wide strips on the left and the right of the image. The result is an "expanded" copy of the original screen:

This uses up the first five blits. If you study the code in the InitBlit() routine in LIFER.C, you see how the bitplane pointers, modulus, height, and width variables all work together to generate the desired blits. Since a plain "vanilla" copy from source A to destination D is wanted, the MINTERM is A_TO_D, and USEB and USEC are set to zero for these blits.

When we calculate the next Life generation, we are assured every cell in the original 320 x 200 screen—including those on the edges—will have valid neighbors on all four sides in the expanded version. We are ready for the nine blits that actually calculate the next generation.

•AC•

GRIDIRON!

WINNER!
"BEST OF 1987"
—Commodore Magazine

WARNING: Independent test results have determined that this game is habit forming.

Experience it!

Fourth and goal on the two. Ten seconds left and you're down by five. The roar of the crowd is deafening as you scan the defense. Something's not right. You call an audible...that special play you've been saving for a moment like this. You fake a hand off to the fullback up the middle and drop back to pass, but nobody's open. You roll left, pursued by a 270 pound defensive lineman. Finally, the fullback breaks clear in the end zone. You squeeze a bullet between two defenders as your fullback cuts back to make the catch. The scoreboard lights up as the crowd goes wild...TOUCHDOWN!!

This is **GRIDIRON!**™, the most realistic football simulation ever developed for a micro-computer. **GRIDIRON!** is **TACKLE** football played in real-time. Call a play from the Playbook, or design your own using our *Play Creation Utility*. You control the action with five levels of play, real time graphics, digitized stereo sound and an unprecedented level of game interaction.

Experience the game that's winning rave reviews from players and critics alike. **GRIDIRON!**, the ultimate football simulator designed to unleash the power of your Amiga. Nothing else comes close.

Requires 512K. For one or two players. Soon to be available for the Atari ST.

Amiga is a trademark of Commodore-Amiga, Inc. © 1986 Bethesda Softworks

To order, call 1-800-245-4525



ELECTRONIC ARTS®

9208 Burning Tree Road, Bethesda, Maryland 20817 (301) 469-7061



BETHESDA
SOFTWORKS

Christmas Fantasy



SYMPHONY MUSIC VIDEO

SYMPHONY MUSIC VIDEO continuously displays pictures and plays music. At the end of a selection, another picture and music piece is loaded. The music and pictures are all in IFF format so you can modify ours with your favorite Paint or Music program or use your own music and pictures and create your own MUSIC VIDEO. You can even have your MIDI synthesizer play the music.

The MUSIC VIDEO that is included is perfect for the Christmas Season. Traditional Christmas scenes and music are continuously played. Fun to listen to and watch. Adds to the best of seasons. \$24.95

Symphony Songs

Ready to Play Music For Deluxe Music, Music Studio, Sonix

A library of nearly 1,000 music masterpieces ready to play with your favorite music program. All selections are in both IFF and MUSIC STUDIO format. Space does not allow listing all songs in each volume, however, a few titles, the number of songs, and the total playing time is given.

Complete list of songs \$3.95.

Each volume is \$24.95 and includes the complete list.

BEATLES Part 1

Vol 15 (21 Pieces 40 Min)
Let It Be, Yesterday, Eleanor Rigby, When I'm 64, . . .

BEATLES Part 2

Vol 40 (15 Pieces 40 Min)
Magical Mystery Tour, Lucy In The Sky With Diamonds, Penny Lane, . . .

CLASSICAL Part 1

Vol 27 (18 Pieces 40 Min)
Prelude #1, Moonlight Sonata 1st and 2nd Movement, . . .

CLASSICAL Part 2

Vol 34 (13 Pieces 40 Min)
Sonata In C Major, Jesus Joy Of Man's Desire, . . .

CLASSICAL Part 3

Vol 31 (14 Pieces 35 Min)
1st Piano Concerto, Polonaise Sonata In C Major, Etude #3, . . .

CLASSICAL Part 4 (Bach)

Vol 35 (22 pieces 30 Min)
Two Part Invention #1, Three Part Invention #6, Prelude and Fugue 1, . . .

CLASSICAL Part 5 (Bach/Clementi)

Vol 46 (24 Pieces 50 Min)
Choral #1, Sonata #1, Theme and 11 Variations From The 2nd Sonata, . . .

BEETHOVEN, BROADWAY, & BLUES

Vol 38 (15 Pieces 40 Min)
2nd Movement Of The Pathétique Sonata, Minuet In G, Fuer Elise, . . .

COUNTRY CLASSICS Part 1

Vol 41 (15 Pieces 45 Min)
Thank God I'm a Country Boy, Act Naturally, . . .

ROCK Part 1

Vol 32 (19 Pieces 50 Min)
AXEL F, Eye Of The Tiger, Both Sides Now, . . .

ROCK Part 2

Vol 16 (20 Pieces 40 Min)
Georgy Girl, Guantanamera, Theme From "Love Story," Cherish, . . .

80's GREATEST

Vol 24 (15 Pieces 50 Min)
Hill Street Blues Theme, Chariots Of Fire Theme, Dynasty Theme, . . .

70's GREATEST

Vol 12 (19 Pieces 45 Min)
Tie A Yellow Ribbon On The Old Oak Tree, We've Only Just Begun, . . .

60's GREATEST

Vol 13 (19 Pieces 45 Min)
Windy, By The Time I Get To Phoenix, Come Saturday Morning, . . .

GOLD & PLATINUM HITS

Vol 45 (19 Pieces 60 Min)
Thriller, 99 Luft Balloons, California Girls, . . .

KENNY RODGERS HITS

Vol 39 (12 Pieces 45 Min)
Lady, Ruby, She Believes In Me, The Gambler, . . .

BILLY JOEL GREATEST HITS

Vol 43 (17 Pieces 65 Min)
Piano Man, Say Goodbye To Hollywood, Only The Good Die Young, . . .

COUNTRY CLASSICS Part 2

Vol 42 (13 Pieces 50 Min)
Ode To Billy Joe, Me and Bobby McGee, Country Roads, . . .

TV THEMES

Vol 37 (21 Pieces 35 Min)
Hill Street Blues, St. Elsewhere Theme, Masterpiece Theater Theme, . . .

MOVIE THEMES

Vol 19 (16 Pieces 40 Min)
MASH Theme, The Rose, Can You Read My Mind (Superman), . . .

BROADWAY'S THEMES

Vol 47 (25 Pieces 65 Min)
The Last Supper, Dr. Doolittle, The Old Dope Peddler, . . .

CHURCH MUSIC

Vol 23 (26 Piece 50 Min)
Amazing Grace, What A Friend We Have In Jesus, . . .

BARBERSHOP

Vol 22 (22 Pieces 45 Min)
Hello Dolly, Put On a Happy Face, Hey Look Me Over, . . .

RICHARD RODGERS SONGBOOK

Vol 18 (19 Pieces 40 Min)
Climb Every Mountain, DO-RE-MI, The Sound Of Music, . . .

NOSTALGIA

Vol 17 (22 Pieces 45 Min)
Let Me Call You Sweetheart, Ain't Misbehavin', On The Goodship Lollipop, . . .

CHRISTMAS

Vol 36 (24 pieces 50 Min)
O Little Town Of Bethlehem, Let It Snow, March Of The Toys, . . .

POLKA PARTY

Vol 33 (17 Pieces 40 Min)
Happy Polka, Pizzicato Polka, Betty Polka, . . .

We accept CASH, CHECK, COD, VISA and MASTER CARD orders.
Shipping and handling US and Canada . . . \$3.00
Shipping and handling outside the US and Canada . . . \$5.00
COD charge . . . \$2.00
Illinois residents add 6 1/4% sales tax.



Speech Systems

38W255 DEERPATH ROAD
BATAVIA, ILLINOIS 60510
(312) 879-6880

AmigaNotes

Digital Music Generation

by Rick Rae
CIS (76703,4253)

The Amiga's software can be viewed as a series of layers: WorkBench and the CLI "sit on top of" Intuition and AmigaDOS, which in turn rely on the Exec routines, and so on. Digital music generation can be broken down in the same way: one layer processes a sequence of instructions—notes, instrument changes, and so on; another layer controls the creation of the various timbres used; another handles the actual tone generation. This month we'll take a look at the lowest level, the actual tone generation process.

In a digital audio system, sound is created by generating a series of values at specified intervals. These values are converted to voltages by a Digital to Analog Converter (DAC), then passed through a lowpass filter to remove unwanted remnants of the digital signal generation process.

The sequence of values determines the timbre, or tonal quality, of the sound. For example, if we generate a set of high values alternating with a set of low values, the result is a square wave. Devoid of even harmonics, a square wave has a hollow sound reminiscent of a clarinet. Repeating a set of continually rising values, on the other hand, results in a rising sawtooth; a true sawtooth contains all harmonics and has a sharp timbre distantly related to bowed strings.

No matter what waveform we want to produce, there are two ways to generate the signals: real time and pre-calculated.

REAL TIME SIGNAL GENERATION

The most obvious way to generate these signals is to calculate them in real time. The CPU, or dedicated hardware, performs the necessary calculations during the interval between samples, then sends the value to the DAC at the appropriate time.

This method has one serious problem: overhead. For anything beyond the simplest waveforms, a CPU like the 68000 simply can't calculate the samples quickly enough. What we need is an approach which insulates the CPU from the time limitations of real time generation.

PRE-CALCULATED SIGNAL GENERATION

One solution is to do the calculations before we need them. Hal Chamberlin was one of the first microcomputer enthusiasts to experiment with this approach. He knew of several methods which could produce remarkably realistic sound, but most required more CPU horsepower than was available in microcomputers at that time. In an experiment he used one of these methods to generate organ tones. Hal gave the CPU as much time as it needed to calculate each value, allowing the program to write the results slowly to mass storage. Once the calculations were completed, a much simpler program could spool each value from disk to the DAC, completing the process.

TABLE LOOKUP

A more manageable approach is the table lookup method. Instead of computing an entire composition, the program calculates one or more cycles of a waveform and places it in a table in memory. The playback program moves a pointer through this table, retrieving values and sending them to the DAC. When the pointer reaches the end of the table, it loops back to the beginning. By changing the number of values in the table, we can change the pitch being generated. If we hold the table size constant, there are two other ways to vary pitch. One method is to change the rate at which the pointer is advanced; the other is to advance the pointer by values other than one.

Amiga sound software normally uses a hybrid approach, combining two of these techniques. To select pitches within one octave, we vary the rate at which the pointer is advanced. To select the octave, we change the table size. (Each octave has its own table, and each table has half as many entries as the one for the octave below it.)

The Amiga, a very flexible system, can utilize all of the sound generation methods mentioned so far. The sound hardware can be set up to loop continually through a waveform table in memory, automatically advancing the pointer and generating values so the CPU can proceed with other business.

(continued)

Or the table can be made very large and the looping disabled, allowing a complete sound sample to be scanned. The special hardware can even be bypassed, allowing the CPU to generate values in real time. Most programs which turn the Amiga into a musical instrument use a variant of the table lookup scheme.

SETTING THE TABLE

Now that we've examined the mechanics of getting the sound out of the audio jacks, let's back up a level: How do we set up the table and calculate the real time values? Here we have a huge number of approaches available. This is an area where experimentation is still valuable. The rest of this column will describe some of the more popular methods, as well as a few which are new and intriguing.

SAMPLING

The sampling method is probably most familiar to Amiga owners. Here we digitize a live or recorded sound, converting it into a string of digital values which become the waveform table. Looping a single cycle of a digitized sound results in a static, synthetic tone which may retain some aspects of the original. A much longer sample, perhaps several seconds in length, is more commonly used. A loop is often created from the end of the table back to somewhere in the middle, which allows the tone to be held as long as is necessary. Percussion sounds, such as drums or staccato tones, are not looped. The main disadvantage of sampling is that it is difficult to radically modify a sampled sound. If you start with a vibrato trumpet note, you're pretty much stuck with a vibrato trumpet note; you can't tweek a few parameters and turn it into a choir.

FREEHAND DRAWING

One or two of the audio packages available for the Amiga allow you to draw a waveform directly for playback. Although this is useful in repairing damaged samples (removing record pops and so forth), it is generally a hit-and-miss approach to creating original sounds. There is no obvious, clear-cut relationship between what a waveform LOOKS like and what it SOUNDS like. Two waveforms which look very different can sound nearly identical, while two waveforms which look very similar can sound very different. The only rules we can apply are very general: sharp edges add brightness (high order harmonics), narrow pulses tend to be thin sounding, and so on. But I won't deny that it's fun to experiment.

ARITHMETIC COMPUTATION

Although all of the techniques from this point on compute waveforms arguably, I'm referring here to a class of algorithms which are very straightforward and can usually be expressed with only a few lines of code.

Filling a table with any of the traditional analog synthesizer waveforms can be done in this manner. A sawtooth, for example, can be generated with the following C code fragment:

```
while (TRUE)
{ if
  (value > MAX) value = -1;
  value += 1;
}
```

This snippet simply adds one to "value" continually until it reaches a preset limit, and then it is reset to zero. If we set MAX to 2π and insert a SIN() function, we can generate a sine wave. Triangle waves, square waves, and pulse waveforms are equally easy to create.

If we placed an output statement in any of these routines, we would generate a continuous waveform, albeit far below any usable frequency. By running through the loop only once and sending the results to memory, we can quickly fill a waveform table with any simply computed waveform.

The problem with simple waveforms of this sort is that they sound, well, simple. Boring might be a better word; without some sort of change to entice it, the ear/brain quickly loses interest. But changing the algorithms to add variety also makes them more complex, eliminating the method's original advantage of simplicity.

ADDITIVE SYNTHESIS

A sine wave is the simplest waveform possible; it is a single, pure tone with no attributes other than amplitude and frequency (and phase, if we measure it with reference to something else). By using sine waves as building blocks, we can build complex tones.

Both the Synergy synthesizer and the Yamaha DX series support additive synthesis. (The Yamaha DX uses other methods as well.) Music Studio was the first Amiga music package to use this approach.

Theoretically, any sound can be created or mimicked using additive synthesis, given enough sine waves. The problem with this technique is that it takes a lot of work to create a sound. As a sound creator, you are given a large number of variables to work with, and it takes quite a bit of time to get everything "just so." Anyone who has tried to reproduce an acoustic instrument using Music Studio knows exactly what I mean.

RESYNTHESIS

Resynthesis, one of my pet approaches, combines Fourier analysis and additive synthesis.

Fourier analysis is the "flip side" of additive synthesis; instead of combining sine waves to create a final product, Fourier analysis breaks a complex waveform down into its component parts.

With resynthesis, we analyze a recording of an existing sound, giving us a set of sine wave frequencies and amplitudes. To use this approach on the Amiga, start with a harmonic-rich waveform like a sawtooth, then use the filter to remove some of the upper order harmonics. This technique can be used to create credible string and brass sounds with a minimum of effort. Unfortunately, the types of sounds this method can produce are limited; realistic acoustic pianos and plucked strings, for example, are particularly difficult to recreate. (Sonix bypasses this problem by giving you several synthesis methods, including freehand drawing, which you can combine to create the final product.)

FM SYNTHESIS

John Chowning (the originator of the FM synthesis concept) and Yamaha (responsible for the commercialization thereof) have certainly shown FM synthesis to be a valid approach. Most people involved in the making of music have heard of the Yamaha DX7, the most successful FM synthesizer of all time.

With FM synthesizing, it is possible to create beautifully complex and realistic sounds. FM synthesis is remarkably adept at simulating plucked strings, struck tines, gongs, and bells. Without some tricks to add "texture," an FM string section can tend to sound a bit thin, but in general FM is a powerful technique, providing an excellent return for the time spent fiddling with the parameters.

SPIRAL SYNTHESIS

Spiral synthesis is one of our new "toys," invented only recently. Con-

ceptually, spiral synthesis looks at the waveform "end on." It considers it a two-dimensional projection of a three-dimensional entity, in which a sine wave represents a spiral. This approach opens up a new viewpoint, which is always valuable in synthesis experiments.

That'll do it for now. Until next month...

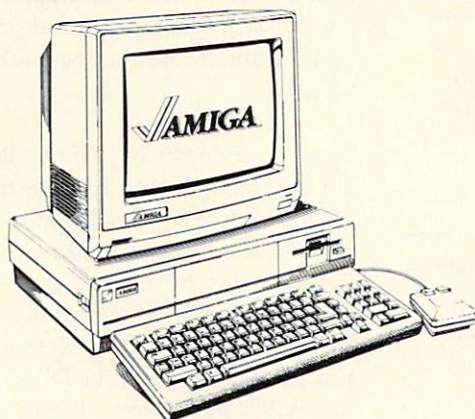


•AC•



ProtoCall

The Next Step in AmigaTM Telecommunications



- Supports All Standard Protocols and Terminal Types
- Very Fast, a Must for a MultiTasking Computer
- *Watch Me* — ProtoCall Will Watch Your Every Command and Write a Script For You, NO PROGRAMMING!
- *Quick Access* Phone Book and Menus
- MultiChannel Transfers Supported
- Interactive Help
- Specifically Designed for the Amiga Computer

Only 49.95!

ProtoCall, The Rest Are History.

Amiga is a Trademark of Commodore-Amiga, Inc.



Data Solutions
 P.O. Box 87190
 Canton, MI 48187-0190
 (313) 397-2889

Animation for C Rookies

Part IV—Out Of Sequence

by Michael Swinger

If you have done any programming on the Amiga, you have probably realized that linked lists are at the heart of many system routines. Whether you are creating menus or windows, or using the built-in animation routines, linked lists are there. You have also certainly found that the lists are fragile—any error at all brings the system crashing down. If your animation programs compile successfully, and the screens and windows and colors all appear when the program is run, but no Bobs are visible, you have probably done something silly (such as pointing an AnimComp at the wrong Bob). Such a trivial mistake prevents the animation and Gel lists from initializing.

Linked lists are also ill-humored. They do not like to be tampered with once created. You can adjust timers, velocities, and screen coordinates in the AnimObs and AnimComps with impunity, but if you try to change the sequence pointers, you better be careful. You might like to prepare several different animation sequences in your programs using the same Bobs, for example.

The secret is to create the list of AnimComps as a strict sequence, from the first to the last, regardless of the order you want to display them in the program. After the animation list has been initialized, you can then switch the sequence pointers by using the extensions provided in the AnimOb and AnimComp structures. (You can add all kinds of things in these extension—calling up sounds, for example.) The following code fragment illustrates one way to program this.

Assume we have created 3 Bobs as AnimComps (Comp1, Comp2, and Comp3), and they are part of a master AnimOb, named Obj1. We want to alternate continuously between the first two views, until the AnimOb reaches a Y screen coordinate of 100. At that point, we want to switch to the third view. The AnimCRoutine in Comp2 is checked every time this view is the current view. If a pointer to our own routine is present, our routine is called. The routine below checks the Y coordinate of the controlling AnimOb.

If the coordinate is less than 100, the routine switches the sequence pointers in Comp2 to redisplay the first view as the next view in the sequence. When the Y coordinate is greater than 100, the routine allows the sequence to progress from second to third view (as linked originally). The Timer and TimeSet parameters are both set to 0, so the third view never changes. Since the view never changes, the sequence pointers in this unchanging view can point to any other AnimComp in the same list, but cannot point at themselves. (Don't write something like Comp2.NextSeq=&Comp2, or Comp2.NextSeq=NULL.)

The last AnimComp in a sequence that does not repeat must have its timers set to 0. You can, of course, add a second sequence of views and more AnimComps to the list, if the list of views was first created in a strict numeric sequence.

```
Comp1.NextSeq=&Comp2;  
Comp1.PrevSeq=&Comp2;  
Comp1.AnimCRoutine=NULL;
```

```
Comp2.NextSeq=&Comp3;  
Comp2.PrevSeq=&Comp1;  
Comp2.AnimCRoutine=&MyRoutine;
```

```
Comp3.Timer=0;  
Comp3.TimeSet=0;  
Comp3.NextSeq=&Comp1;  
Comp3.PrevSeq=&Comp2;
```

```
MyRoutine ()  
{  
    if (Obj1.AnY < (100*64))  
        Comp2.NextSeq=&Comp1;  
    else Comp2.NextSeq=&Comp3;  
    return ();  
}
```

Double-buffering is memory intensive, and the structures necessary to turn the Bobs into Animation Objects consume even more precious Chip memory. Rumors persist about upgraded graphics chips for the A500 and A2000. (I hope can it somehow be fit into the 1000.) Until they arrive (which may or may not be within the lifetime of this planet), there are ways to conserve memory.

The last AnimComp in a non-repeating sequence can be removed from the Gel list by changing the NextOb and PrevOb flags in the AnimOb structures. It can also be moved off screen, or left in place in the display. If this last Bob/AnimComp is never going to move again, it need not be double buffered.

I know the RKM says, "If one Bob is double-buffered, then all Bobs must be double-buffered." That's not so. Just don't allocate a Save Buffer or a Double-Buffer for the Bob that is not double-buffered. Eliminate the SAVEBACK flag in the Bob structure and place a NULL in the DBuffer and SaveBuffer parameters. If you are not using the system collision detection routines, you can also eliminate the Border Line buffer and place a NULL in the VSprite structure.

•AC•

Manage Your Data... Not Your Database.

Storing data is one thing.
Finding it again is another.

Introducing...

the MICROFICHE Filer

For the first time you can scan your data using intuitive mouse operations rather than struggle with obscure keyboard commands. With our unique software, your data is organized as on a microfiche—as a condensed two-dimensional sheet of text and pictures over which you drag a “magnifying glass” to locate your information.

At last—a Database that fully utilizes the Graphics capabilities of your Amiga!

Unlike inferior database programs, the Microfiche Filer allows you to store pictures of any size. You can also display several pictures *simultaneously* and mix pictures with text.

Some Additional Features:

POWER
FLEXIBILITY
CONFIGURABILITY
RELIABILITY

Full-powered sorting, selecting, and printing.
Fields may be added or deleted at any time.
Adaptable forms with which to display data.
Not copy protected.

You can now own the MICROFICHE FILER
for just **\$99.** (plus shipping)

SHIPPED SAME DAY when ordered
with MasterCard or VISA!

CALL
TOLL
FREE

800-527-7014

In MA (617) 877-1266
to order or for location of nearest dealer.



SOFTWARE VISION Inc.
26 FOREST ROAD, FRAMINGHAM, MA 01701



We Focus

In the growing video market one company leads the way in top-quality, low-cost, computer effects software. Aegis. You'll find our products working at every level in the video field. Professionals in television, video production, cable TV, industrial and government video, college, schools—you name it, we're there.



At MetaVision

When Theo Mayer and Peter Inova began work on a video project for Universal Studios tours they decided to try out an Amiga for a particular effect they wanted. It worked perfectly, and they saved thousands of dollars in post production fees. Next, they put Aegis products to work producing animations for a McGraw-Hill home

We're there because our products provide flexible technology that sparks the imagination. Technology that gets the job done.

You'll find us at Prism Graphics
For Jeff Bruette the Amiga spells success. He uses Aegis programs like VideoScape 3D, VideoTitler, and Animator to help him produce graphics and animations for television programs like *Max Headroom* and *Secrets and Mysteries*.



video. They were convinced, and the Amiga with Aegis software became an integral part of the studio.

At Master Communications

The publishers of

the popular *Board Sailing Year* annual video put Aegis software to work for their corporate clients. An Amiga 2000 is part of their video production house and they use products like VideoScape 3D, Animator and AudioMaster for special effects and logos.

And at Pixelight

Independent Amiga artist, Nick Poliko, uses Aegis products to create graphics and animations for such clients as CommCorp, Purolator, and Union Gas. They use his artwork for everything from public relations videos to attention getters at trade shows.

With Products for Graphics, Animation

We handle metamorphic, cel, and color cycling animation with Aegis Animator. Animations are created and edited in



an interactive environment allowing you to watch your animation as you build it. Winner of the 1986 CES Award of Excellence, Animator has already found thousands of uses around the world. Included in the \$139.95 price is the Images paint system.

Award winning in its own right Images is a lot of paint program for a small price. With color cycling, mirrors, air brush, pantograph, gradient fill, over 40

Aegis Animator, Images, VideoScape 3D, AudioMaster, Sonix, VideoTitler, and Impact are trademarks of Aegis Development, Inc.

ANIM is a trademark of Sparta/Aegis Development, Inc. Amiga is a registered trademark of Commodore-Amiga, Inc.

The AFB Camcorder is a product of Sony Corporation and its use in this ad is not an endorsement by Sony of Aegis products.

on Video

other features, and a price of \$39.95, Images has the best price/performance ratio in paint packages.

Business graphics are a snap with Impact! Pie, bar, line, area, symbol, and scatter-gram charts are all easily created with every element of the chart from labels to axes available for editing. You also get a slide show generator with eight different dissolves and wipes. All for \$89.95.

For the advanced animator VideoScope 3D provides an environment rich in 3-dimensional capabilities. Object motion and metamorphosis, camera motion, light sources, IFF foregrounds and backgrounds, and the ability to create animations in the ANIM format are just a few of the features that make the \$199.95 price tag a great buy.

For titling you can't beat Aegis VideoTitrer. It supports all of the Amiga fonts as well as its own polytext fonts, works in four different resolutions and uses overscan. It has 20 different styles, works with IFF, uses halfbrite if available, and supports the ANIM format. Included in the amazing \$149.95 price is a slideshow generator that can mix ANIM animations with slides.



And Sound

Our video effects don't stop with great visuals. Programs like Sonix and AudioMaster take on the world of music and sound as well. Winner of a CES 1987 Award of Merit, Sonix lets you create your own instruments, compose music, and work with MIDI instruments. All for \$79.95.

We back up Sonix with AudioMaster, a digital sampling and editing program. It features interactive editing of the waveform and effects like echo, reverse, and low pass filtering. It also makes use of expansion memory (up to 9.5 MB) for extra long samples. Just \$59.95.

For more information and the dealer nearest you: (213) 392-9972 or to order direct: 1-800-345-9871.



AEGIS

2210 Wilshire Blvd., Suite 277
Santa Monica, CA 90403

The Hack & Slay-Team presents:



Watch out
for
Snake Design!



We offer you:

- * In every game a bonus point.
- * Collect the points and earn:

for 3 points a Snake button.
for 10 points an arcade game for free.

See the games in our next advertisement.

Only software from Snake Design offers you this features!!!

A new Software Generation!

Witchcraft

49,⁹⁵

- Bored of destroying evil wizards? We get a more intelligent plot!
- Lots of mindboggling logical puzzles, riddles and all those monsters lurking in the dungeons!
- More monster graphics than in any other fantasy game!



Emetic Skimmer

34,⁹⁵

- Steer your spacecraft through endless tunnels and deep into the electronic caves of the supercomputers security system.
- Avoid hazards, destroy bases and finally find the deadly super-computer-complex.
- Featuring two disks full of graphic and action!



Garrison

49,⁹⁵

- Fight through the 128 giant rooms of the castle.
- Defend yourself against ghosts, magicians, wardens, vampires and phantoms.
- Can you rescue the lovely princess?
- Two player option, five characters to choose. Simply the best graphics game!



Spaceport

39,⁹⁵

- Prove your skill steering the helicopter.
- Rescue the people in the mines.
- Destroy enemies and walls.
- Find the aliens power station.



Mission Elevator

49,⁹⁵

- Find your way in this strange hotel.
- 64 different floors, different kind of enemies.
- Be fast and shoot the agents, play power, throw the dice, pay beautiful girls!
- Find objects to reach the next floor, but be careful.



Detonator

39,⁹⁵

- Zap, catch and bang through 50 levels of pure fun.
- Digital speech, sound and music.
- Collect extra actions & weapons.
- Scrolling background graphics supports this ultimate game.
- Bonus levels in true 3D, the game in the game!



Available at these addresses:

C. Ltd
723 East Skinner
Wichita, Kansas 67211
Phone: 316 267 3807
Fax: 316 267 0111

American Software Distributors
R. R. 1, Box 290, Building 3
Urbana, Illinois 61801
Phone: 800-225-7941 or
217 643 2050
Fax: 217 643 2049

Impulse Inc.
6870 Shingle Creek Parkway 112
Minneapolis, Minnesota 55430
Phone: 800-328-0184 or
612 566 0221
Fax: 612 566 1822

Mindware International
110 Dunlop St. W. Box 22185
Barrie, Ontario
Canada, L4M5R3



Snake
Design
Software
Phone: 01149-69-7071102
Fax: 01149-69-708525

FormatMaster: the Professional Disk Formatting Engine

by Carl Mann

Perhaps one of the least appreciated programming Amiga languages available is the very first one that executes when the system is first powered-up and booted (after the Kickstart machine code, of course). This is the language which puts the familiar "Use Preferences tool to set date" message on your text screen. This neglected resource is capable of revolutionizing your approach to the Amiga system. It is easy to use, powerful, and enables you to make your machine do complex, repetitive tasks (such as formatting disks) with an absolute minimum wasted motion. This amazing tool is called AmigaDOS "Batch Language."

Batching for Beginners

What exactly is Batch Language? Simply stated, it is a pre-programmed sequence of DOS commands which can be consecutively executed by the host computer. Execution of these commands is controlled by the orderly arrangement of instructions contained in an ordinary ASCII text (or "Script") file. (On other computers, Batch Language is known as Job Control Language, EXEC, or AUTOEXEC.) The AmigaDOS Batch Language elements are contained in the "C" directory of your Workbench system disk, mixed in with the CLI commands. The conventional place for Amiga Batch language "Script" files is the "S" directory of that same Workbench disk.

There is a subtle difference between Batch and CLI commands: Batch language script files can invoke any

and all AmigaDOS commands and options. From CLI, though, it is nearly impossible to find a good use for a hand-typed "IF... NOT... THEN... ELSE... SKIP... ENDIF" condition-testing and branching construct or a "wait for the keypress" routine. Both these techniques and much more can be built into Batch programs.

The very first thing your Amiga does when you insert a Workbench disk is execute a Batch file. This file, named STARTUP-SEQUENCE, resides in the "S" directory of every viable Amiga Workbench disk. Without this file, the Amiga is no different from any other keyboard-driven silicon dinosaur. However, most STARTUP-SEQUENCES are rudimentary beasts at best. As delivered from your friendly neighborhood software supplier, the STARTUP-SEQUENCE generally does little more than remind the operator to set the date. It then may flash the title of the upcoming program onto the screen. Following this, display either the main Workbench executable module (named LoadWB) or the actual application program may be invoked. As a rule, the final act of such a file commits a certain form of suicide with a line that reads, "ENDCLI > NIL:." This statement closes the text window to reveal the familiar Workbench Intuition environment screen.

Very good. The Amiga's all booted up, and I know everything is all right. I can use the mouse to open a window on my Workbench disk and I can even pull down my Workbench menus. Now I want to do some useful work.

I will probably need to use that box of disks, right? Right. Here is where the pain (or fun, if you are a masochistic person) begins.

Building the Perfect Bootable Disk

The difficulty in preparing formatted, bootable Amiga disks is especially severe if (like me) you start your Amiga career without an external disk drive. Swapping the Workbench disk with the target disk and back again is just not fun beyond the first few exchanges. First you click. Then you pull down to "FORMAT." Then you swap. Then you swap again. Then you click again. Then you wait. Then you swap again. Are we all in the same place now? Good! Now let's all open up a CLI window and learn how to do an INSTALL! Ready now, class? Hey, where did everybody go? Oh, forget it.

FormatMaster relieves the frustration a friend and I both experienced with the ordinary ways of preparing Amiga disks for everyday use. You see, we expect our computers to serve our needs, not the other way around. The speech option was originally inspired by my friend David's seemingly "special" needs. (He's blind, you see.) Later, I needed to prepare twenty bootable disks for an upcoming event and wash a double sinkful of dishes at the same time. I then learned the true worth of properly applied speech technology.

(continued)

This project is based on Amiga Workbench 1.2. However, there is a little-known bug in the WB1.2 "FORMAT" command file which makes it bomb out with a failure code of 20. This bomb occurs when FORMAT 1.2 is executed from RAM: by a RAM:-resident Batch file. What's worse, it gobbles up about 30K of memory every time it does so. This causes an eventual GURU upon repeated execution. However, the FORMAT command from Workbench 1.1 works just fine. (Commodore's Software Development liaison insists this nastiness is actually a feature. Unfortunately, she had a hard time explaining the usefulness of this filthy behavior in rational terms.)

To succeed at this project, you will need the following resources:

- An Amiga computer system. (512K is required. One drive is OK.)
- Workbench 1.1 (for its Format command)
- Two fresh backup copies of your Workbench 1.2 disk.
- The ability to start and use ed, the Amiga line editor. (The WB1.2 version of ed is much better than the WB1.1 release!)
- Some skill in CLI commands, or a good file manipulation utility. (I highly recommend ZING! from Meridian Software.)
- The two Batch files, S/STARTUP-SEQUENCE and PREPAREDISK.EXE, appearing in this article. (Both files must be copied *verbatim* and placed in the *proper* places on a Workbench 1.2 disk to produce success!)

The Startup-Sequence File

S/STARTUP-SEQUENCE (Listing One) is a customized, turbo-charged Batch file. It transforms the stock Amiga computer into a factory for formatted, bootable disks. Under directions from this "script" program, the Amiga first sets aside 100K of disk buffer space. This apparently extravagant use of memory enables the system to load the "SAY" command

into RAM storage from disk only once, yet execute it as many times as necessary. This feature, called "caching," is a powerful hidden aspect of WB 1.2.

STARTUP-SEQUENCE then creates an explicit logical path from the boot disk's "Root" directory to its "System" directory. This enables AmigaDOS to locate and execute the "SAY" command from disk. (Without this "path," AmigaDOS will bomb with an "ERROR CODE 205," which means, "I can't find it!")

The next STARTUP-SEQUENCE event is optional. I like to minimize the number of electrons per second streaming from the hot cathode of my expensive high-resolution monitor CRT. (I just don't need a video screen blazing away into empty space while I bag the trash in the kitchen.) To avoid wasted electrons, I include the ScreenBlanker routine that MICROSMITHS bundles with their FastFonts text utility package on most of my Workbench disks. This one is no exception. Omit or comment this line out of the Batch files in this article if you have no such routine in your Amiga library at the moment. Better yet, get FastFonts...

STARTUP-SEQUENCE next copies the "COPY" command file to the AmigaDOS RAM: device and changes the current directory to be RAM:. "COPY" executes directly from RAM:, bringing necessary elements of the FormatMaster system into the RAM: device as directed by the script file. Subsequent file copy and other execution operations take place strictly from RAM:. This serves three purposes. First, "COPY" need no longer be periodically re-loaded from the FormatMaster disk for re-execution. Instead, it executes directly from RAM:. Second, the target files also execute very quickly from RAM: because the continual disk access that AmigaDOS must undergo when executing consecutive command files is now handled electronically. Third,

DF0: becomes available as a full-time production tool for the duration of the formatting session. Just imagine—no more "PLEASE INSERT VOLUME WORKBENCH" system requesters!

FormatMaster's STARTUP-SEQUENCE also uses an undocumented, but genuine feature of the AmigaDOS 1.2 "COPY" command to further reduce startup time. Note the vertical bar characters separating the target file names in the program lines which invoke "COPY." As you can see, the WB1.2 version of "COPY" can duplicate multiple files with ease. It is no longer necessary to execute a separate "COPY" command for each file to be duplicated. Instead, a number of files to be copied are entered as arguments on the Batch file command line with a vertical bar character separating ("delimiting") them.

There are a few limitations to this use of "COPY," but the example is an adequate illustration of acceptable syntax. Just make certain each file in the concatenated list is the same type and in the same directory as its neighbors. WB1.2 "COPY" does not cross filetype, directory, or device boundaries in mid-argument stream. In addition, no more than four or five files may be strung together at one time. (The exact number seems to depend on target file size.) As published, though, the system works very well.

The "QUIET" option of "COPY" is invoked to suppress the reporting that would otherwise take place as "COPY" grinds away. The really critical events in the sequence are announced by the "SAY" command. This eases tracing of the program's execution.

Following its copy work, S/STARTUP-SEQUENCE loads the Workbench 1.2 Intuition environment for later use. Control is then passed to the various command files previously copied to the RAM: device and to another script file, named PREPAREDISK.EXE (via the final "EXECUTE" command).

The Disk Preparation Loop

PREPAREDISK.EXE (Listing Two) is the actual AmigaDOS-based formatter utility. It belongs in the "Root" directory of the FormatMaster system disk. This Batch file performs its task strictly from the Amiga RAM: device. It invokes the AmigaDOS files that S/STARTUP-SEQUENCE previously copied into RAM:. These command files consecutively format the blank target disk (following a tap of the <RETURN> key), make it bootable (with "INSTALL"), and copy your favorite DISK.INFO and TRASHCAN.INFO files onto its surface. This last step makes it a breeze to produce a standardized, custom icon display. (I personally enjoy the appearance of the artfully animated DISK.INFO and TRASHCAN.INFO icons by Greg Tsadilas that arrived on my SCRIBBLE! word processor disk.)

Once PREPAREDISK.EXE finishes processing your target disk, the event is announced in synthesized English. This process is then repeated on drive DF1:. Once DF1: has done its duty, PREPAREDISK.EXE then invokes itself again. "FORMAT," in turn, again waits for the <RETURN> key to be tapped before proceeding. This enables you to eject the freshly prepared disk. Another "raw" unit may then be inserted for processing. As noted in the listing, you can omit the marked portion of this file if you have a single drive Amiga system. If this is the case, you will most certainly appreciate the ease with which FormatMaster enables you to prepare a quantity of disks. Three or more drives? Just repeat the code sequence and change the drive references as appropriate. Ain't science grand?

Building the System

If you are at ease with the Amiga CLI and ed, you can give yourself the gift of continuous Amiga disk formatting by following these steps:

- 1) From CLI, ED S/STARTUP-SEQUENCE on a fresh copy of your Workbench 1.2 disk. Delete everything in sight. Type in Listing One exactly as printed. If you do not have FastFonts, omit or comment out the ScreenBlanker routine line, then exit with <ESC X> to save the file.
- 2) From CLI, ED PREPAREDISK.EXE and copy Listing Two verbatim. Leave out the marked code section if you have a one-drive system, of course. Make certain you save your file to the root directory (top level) of your project disk. (Make it work first; then "tweak" it.)
- 3) Copy the "FORMAT" command from the "C" directory of your Workbench 1.1 disk to the "C" directory of your WB1.2 project disk. Use whatever means are available. The project will not work without this specific file.
- 4) Copy your favorite custom Disk.info and Trashcan.info icon files onto the project disk if you wish.

Using the System

Once you have fulfilled the directions, create a backup copy of *your work*. (I don't know why folks forget to do this!) Label both your newly created disks. Then *protect your work* by opening the Write Protect window on your "Worker" copy of the project. Place your new FormatMaster disk in DF0: and reboot with <CONTROL> <AMIGA> <AMIGA>. DF0: whirrs and gronks for a few moments and you should then be greeted by the title announcement and the "Please remove your boot disk" prompt.

Note that you may easily format your FormatMaster disk by accident. Careful operating habits are important. Eject your boot disk when PREPAREDISK.EXE prompts you to do so. Just in case, though, make certain the write-protect tab window is OPEN before you insert your hard-won work in the first place! (Oh, and please keep a backup copy on file!)

Once the FormatMaster system is executing, you may format as many disks as your budget will support, without anything more demanding than a tap of the <RETURN> key. I hope you agree that FormatMaster is a worthwhile addition to your Amiga toolbox.

This version of FormatMaster does not include a "clean" way to exit. To exit, simply reboot. If you need to use Workbench, just resize the FormatMaster window and proceed as always. (Mind you, one drive will probably be tied up...) You can also pop the disks out of the drive and tap <RETURN>. FORMAT will promptly bomb out and free up the drive. You can always restart FormatMaster by typing EXECUTE RAM:PREPAREDISK.EXE from CLI.

If you wish to avoid ed and receive the FormatMaster Batch files directly, send a SASE (44 cents postage, please!) and a backup copy of your WB1.2 disk. Three dollars will cover my time and incidental expenses (motoring over to the post office, wear and tear on the drives, etc.).

Please feel free to distribute these batch files to other Amiga owners and users. David and I are especially eager to put this product in the hands of sight-impaired people who wish to use their Amigas more effectively. (I reserve the word "vision" to denote an inner quality not in any way connected with eyesight.) We particularly welcome direct correspondence from blind people and their friends. We intend to use our combined skills to put practical Amiga systems in the hands of folks who need speech-based computers, but who are not willing or able to spend thousands (or tens of thousands) of dollars for other existing (read outmoded and quack-voiced) hardware technology. We are certain the Amiga can be the solution for those of us who are presently unable to climb aboard the bus. Kindly direct correspondence to the author care of this magazine.

(continued)

Enjoy FormatMaster! Spread it around! !Viva Amiga!

Sources for products mentioned in this article:

Scribble! MicroSystems Software 12798 W. Forest Hill Blvd. W. Palm Beach, FL 33414 (800) 327-8724 \$99.95	FastFonts MICROSMITHS Box 561 Cambridge, MA 02140 (617) 354-1224 \$34.95
ZING! Meridian Software Box 890408 Houston, TX 77289 (713) 488-2144 \$79.95	

Listing One: Startup

```

*****
;*
;* This startup-sequence goes in the
;* "S" directory of a standard Work-
;* bench 1.2 disk. Its presence is
;* required by the FormatMaster disk
;* preparation system. This file plus
;* the PREPAREDISK.EXE file comprise
;* the FormatMaster Disk Prep System.
;*
;* Filename = S/Startup-Sequence
;*
;* Carl W. Mann
;* 30 Warren Avenue
;* Amesbury, MA 01913
;*
*****
;
Addbuffers df0: 100 ;Reserve 100K of disk buffer space for
;command caching.
path sys:system add ;Make "SAY" command quickly available
run Screenblinker 5 ;Call your own PD or commercial
;screenblinker routine here, or omit.
;
Echo " Disk Preparation Toolkit V2.5"
Echo " By C.W. Mann"
Echo " 8/25/87"
say Super Disk Formatter for the Amiga? Version 2 point 5.
say Now copying needed files to RAM. Eyell be back soon.
copy df0:c/copy to ram: ;First things first!
cd ram: ;Now shift gears & GO!
copy df0:preparedisk.exe to ram:
copy df0:disk.info|trashcan.info to ram: quiet ;Only files
; of same type & in same directory can be
; concatenated as objects for COPY... no need to list them
onscreen.
copy df0:c/echo|install|assign|makedir to ram: quiet
copy df0:c/format|execute to ram: quiet ;Must use Workbench 1.1
; FORMAT command file here...
copy df0:system/say to ram: ;Keep it talking...
loadwb
execute preparedisk.exe

*****
;*
;* PREPAREDISK.EXE disk formatter
;* utility. Configured for two-drive
;* system; modify as necessary. Put
;* this file in the root directory
;* of your FormatMaster system disk.
;* Copy it to and run it from RAM:
;* using my special S/STARTUP-SEQUENCE.
;* (Ditto for the command files it
;* calls.) Enjoy high-speed Amiga disk
;* formatting today! !Viva Amiga!
;*
;* Carl W. Mann

```

```

;* 30 Warren Avenue
;* Amesbury, MA 01913
;*
*****
;
Echo "
Echo "
Echo " Preparedisk V3.00"
Echo " By C. W. Mann"
Echo " 8/26/87"
Echo "
say Please ree move your boot disk.
say Put disks to be formatted in D F 0 and D F 1? Press ree turn to
format D F 0.
format drive df0: name "Empty"
install df0:
copy disk.info|trashcan.info to df0: quiet
makedir df0:trashcan
say Disk in Drive Zero is formatted and bootable.
;
*****
;*
;* Leave out everything from here down *
;* to the next row of asterisks if you *
;* own a single-drive system. *
;*
*****
;
say Now press ree turn to format Drive 1.
format drive df1: name "Empty"
install df1:
copy disk.info|trashcan.info to df1: quiet
makedir df1:trashcan
Say Disk in Drive 1 is formatted and bootable. Next victim!
*****
execute ram:preparedisk.exe

```

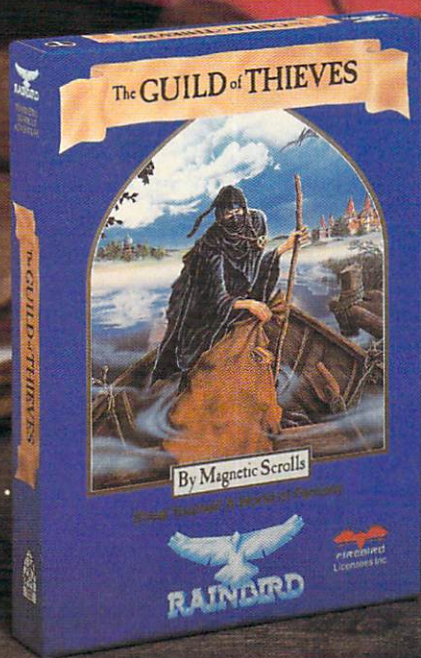
Listing Two: Preparedisk.EXE

```

*****
;*
;* PREPAREDISK.EXE disk formatter
;* utility. Configured for two-drive
;* system; modify as necessary. Put
;* this file in the root directory
;* of your FormatMaster system disk.
;* Copy it to and run it from RAM:
;* using my special S/STARTUP-SEQUENCE.
;* (Ditto for the command files it
;* calls.) Enjoy high-speed Amiga disk
;* formatting today! !Viva Amiga!
;*
;* Carl W. Mann
;* 30 Warren Avenue
;* Amesbury, MA 01913
;*
*****
;
Echo "
Echo "
Echo " Preparedisk V3.00"
Echo " By C. W. Mann"
Echo " 8/26/87"
Echo "
say Please ree move your boot disk.
say Put disks to be formatted in D F 0 and D F 1? Press ree turn to for-
mat D F 0.
format drive df0: name "Empty"
install df0:
copy disk.info|trashcan.info to df0: quiet
makedir df0:trashcan
say Disk in Drive Zero is formatted and bootable.
;
*****
;*
;* Leave out everything from here down *
;* to the next row of asterisks if you *
;* own a single-drive system. *
;*
*****
;
say Now press ree turn to format Drive 1.
format drive df1: name "Empty"
install df1:
copy disk.info|trashcan.info to df1: quiet
makedir df1:trashcan
Say Disk in Drive 1 is formatted and bootable. Next victim!
*****
execute ram:preparedisk.exe

```

•AC•



This is the difference between reading an adventure and living one.

Picture the most exciting text-only Adventure in your software collection—WITH PICTURES!



GUILD OF THIEVES is the long-awaited sequel to **THE PAWN**, which came sizzling onto the software scene to awards and accolades. Both are available now from Firebird.

These stunning Adventures, completely different from any game you've ever played, seduce the sight with

their breathtaking illustrations and engage the intellect with a truly revolutionary text-handling system. The most sophisticated parser on the market lets you input complicated sentences and interact with a whole cast of fascinating characters in the mythical kingdom of Kerovnia.

GUILD OF THIEVES and **PAWN**, Computer Entertainment Adventure of the Year, come to you from Firebird, publishers of world-famous leading edge games under the Rainbird label.

You'll find the Firebird logo on other addictive Adventures too, as well as on absorbing Strategies, realistic Simulations, and fast-paced action Arcade games.

Firebird brings the best in entertainment software to those who enjoy a whole range of interactive excitement. We'll prove to you that you don't have to keep switching brands to satisfy your obsession for challenge!



“The First Full Line In Software”

Firebird Licensees, Inc.
P.O. Box 49, Ramsey, NJ 07446
(201) 444-5700

Available for C-64, Amiga, Atari 520ST, Atari 800/130, Macintosh, IBM and compatibles. A "text-only" version is available for Apple II computers at \$39.95.

Firebird, and the Firebird logo are registered trademarks of Firebird Licensees, Inc.



* (520ST Graphics)

IBM is a registered trademark of International Business Machines Corporation. Amiga and Commodore 64 are registered trademarks of Commodore Business Machines, Inc. Macintosh and Apple II are registered trademarks of Apple Computer, Inc. 520ST is a registered trademark of Atari Corporation.

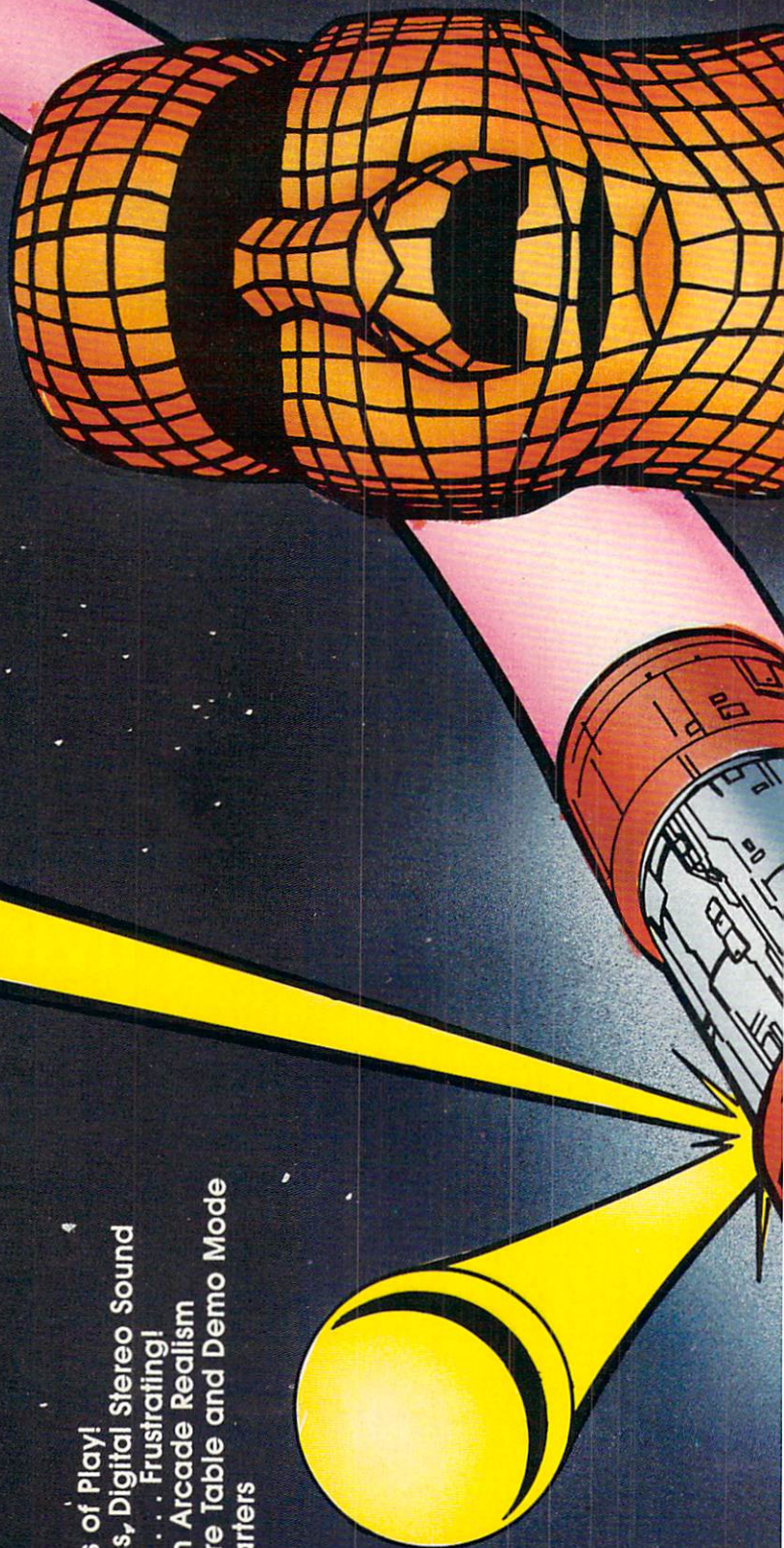
TM

ARCADE

THE #1 ARCADE BLOCK-BUSTER!

- 33 Sizzling Levels of Play!
- Dazzling Graphics, Digital Stereo Sound
- Fast! Fun! Furious! ... Frustrating!
- A Breakthrough in Arcade Realism
- Arcade High Score Table and Demo Mode
- Never Needs Quarters

ONLY \$49.95 • !



Licensed from
TAITO AMERICA CORPORATION



SIMPLY THE BEST AMIGA GAME EVER!

Taito and Arkanoid are trademarks of Taito America Corporation. ©1987 Taito America Corporation. All rights reserved.



"Marauder II is an excellent utility at a bargain price."

Bob Ryan-Amiga World

"Every Amiga owner should have a copy . . ."

Bruce Webster-BYTE

"If you own an Amiga and don't have Marauder II from Discovery, you don't have one of the best tools an Amiga owner can have." -INFO Magazine

The best copier for Amiga™ at any price! Still only **\$39.95 ***

Our new **DX SERIES™** provides the power and features of the world renowned Hewlett-Packard advanced programmable calculators at a fraction of the price. The **DX-16™** is a perfect on screen replica of the HP-16C™, the ultimate programmer's calculator. The **DX-11™** mirrors the HP-11C™, an excellent choice for students, engineers and general purpose calculations. Both are included to make this package a double value. Both are fully programmable and fully multi-tasking power tools. Complete tutorials and excellent documentation will get you started quickly. **\$49.95 ***

NEW! DX SERIES

NEW! AMNIX

AMNIX™ delivers the finest features of the famous UNIX™ C-Shell. Free yourself from constant disk swapping. Over 40 high-speed resident commands make their disk-based Amiga DOS counterparts obsolete. **AMNIX™** also provides extended batch files, command line history and editing and environment variables. It's easy to learn and superbly documented. A "must have" for Amiga power users!

\$49.95 *

LIMITED TIME OFFER!
ORDER DIRECT AND GET
\$5.00 OFF
EACH PRODUCT YOU BUY
PLUS



ABSOLUTELY FREE!!
TO ORDER CALL TOLL FREE
1-800-34-AMIGA
or (301) 268-9877

or send check, cash or

VISA or MasterCard numbers to:
Discovery Software International Inc.
163 Conduit Street
Annapolis, MD 21401

*Order now for \$5 introductory discount!
(Federal Express shipping in continental U.S. only.)
OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Forth!

Memory in the Amiga

by Jon Bryan

Memory in the Amiga, as you no doubt know, is divided into two basic types, CHIP and FAST. CHIP memory is located in the first 512K of the 68000's address space and is shared between the CPU and the Graphics, Audio, and DMA coprocessors which make the Amiga so powerful. The remainder of the 68000's address space is considered FAST memory. The main difference between the two types of memory is, while the 68000 has complete, unobstructed access to FAST memory, it must share CHIP memory with the coprocessors. Usually CHIP memory access is interleaved, but when the coprocessors need extra time, they can steal it from the CPU.

Since it is generally better to run programs from FAST memory (for speed and conservation of CHIP memory), the operating system puts programs in FAST memory by default, unless told otherwise. Unfortunately, a program which contains pictures or sound data does not run properly if it is loaded entirely into FAST memory. The coprocessors cannot access that data to display it or play it back.

For this reason, the Amiga supports a technique called "scatter loading." A program may be saved to disk so the CHIP "hunks" are sifted out from the FAST hunks when it is loaded into the Amiga's memory. Many early Amiga programs (my own included) failed to make use of this capability and do not run properly on a machine with more than 512K of memory.

The latest release of Multi-Forth (Version 1.21) really simplifies the process of properly "turnkeying" a program to avoid this problem. Just follow this procedure: Use GET.MEMORY to ask the system for a block of RAM. Save the handle and flag the word which references the handle with IN.HEAP. This step allows FORGET to return the memory to the system and TURNKEY to save the hunks along with the program, so they will be restored when the application is run. I've rewritten the hardware sprite defining tools which were presented several issues back to demonstrate the technique. They worked fine on my 512K Amiga when I wrote them, but adding Michigan Software's one megabyte Insider caused all the sprites to disappear. I was also unable to resist the opportunity to incorporate a few other improvements in the code. This version works on

releases of Multi-Forth prior to Version 1.21, but those of you who have neglected to upgrade must do a little more work when you turnkey the program.

Before describing the revisions made to support machines with FAST RAM, I'd like to briefly cover some other minor changes in the code. For instance, the original version of the word "SpriteLine" used the following definitions:

"SpriteLine" used the following definitions:

```
256 CONSTANT ScanBufSize
CREATE ScanBuf ScanBufSize ALLOT
: SpriteLine (—addr1\addr2)
ScanBuf ScanBufSize INFILE @ READ.TEXT 1- ( trim delim )
ScanBuf + DUP 16 - ;
```

Its purpose was to parse 16 characters from a file which could then be interpreted and separated into bit planes. I was never very happy with the clumsiness of the definition and the need to create a buffer to hold the characters. The obvious choice for the job, "Word," can't get past the linefeed character which delimits each line in a text file, through. I eventually discovered the Multi-Forth word "SCAN" in the glossary, and used it to build a modified version of Word which gives the desired behavior.

```
: WORD1 ( delimiter—addr )
scan error" premature delimiter "
pocket bl over count + c! ( blank delimit the string ) ;
```

"Word1" automatically crosses and keeps track of line boundaries. "Blank" delimits the string (83-Standard behavior for Word, by the way), leaving the address of the counted string on the stack. "Word" is potentially extremely useful and has a permanent place in my toolbox. In fact, I can't understand why Creative Solutions didn't define "Word" to behave this way. The SpriteLine definitions now become much more readable:

```
: SpriteLine (—addr1\addr2) \ Indices for a DO...LOOP
bl word1 dup c@ 16 = not
error" must be 16 characters! " \ a little error check
1+ dup 16+ swap ;
```


The change in SpriteLine is cosmetic and doesn't extend beyond itself. A more important change—one dealing with the proper use of memory—appears in the updated definition of "Sprite." "Sprite" is the word which actually creates the binary image of a "simple" sprite in memory. The old version of Sprite used "ALLOT" to allocate space for the image in the dictionary. It worked fine on a 512K machine because it was ALLOTting CHIP memory. If the dictionary is in FAST RAM, however, "ALLOT" creates a serious problem. The solution involves using the word "GET.MEMORY" with the right parameters to ask the operating system for a hunk of CHIP memory to hold the image:

```
: Sprite ( name ( height—)
SimpleSize CHIP CLEAR or get~emory \ returns a handle
dup 0= error" couldn't get CHIP memory for sprite "
dup CREATE , \ save the handle (behaves like a VARIABLE)
in.heap \ and flag the new word
@ \ handle points to a CHIP address
4+ swap \ first 4 bytes are for system use
SimplePlanes ;
```

If successful, GET.MEMORY returns a "handle" which contains the address of the allocated memory. Otherwise, it returns zero and an error is generated. Sprite CREATES a new word and flags it using "IN.HEAP" to indicate that it has a heap handle in its parameter field. That handle points to the block of CHIP memory where the sprite's binary data is stored. The process of flagging words with IN.HEAP allows "FORGET" to return the memory associated with those words to the heap, and allows "TURNKEY" to save the hunks along with the rest of the file. It is pretty unlikely that the hunks will be reloaded in the same place, so Multi-Forth also takes care of updating the handles when a turnkeyed program is loaded.

Equivalent changes were made in the word "Attached," which compiles "attached" (16-color) sprites. "Attached" is a bit more complicated, but the principle is the same. An attached sprite consists of two simple sprites, with one sprite "attached" to the other to quadruple the number of colors. "Attached" asks for a memory hunk large enough to hold both images.

"Sprite" and "Attached" create binary sprite images in CHIP memory and let you know where those images are stored, but one more level of sophistication is required to actually display the sprite. In Forth, you can create a defining word to do all the work for you, as I have done in the word "MakeAttached." It creates a sprite image and gives it a name which, when executed, displays the sprite:

```
: MakeAttached ( name ( height—)
Attached \ create the image and give it a name
DOES> (—) \ specify its behavior
@@ \ fetch the contents of the handle
\ (the CHIP address of the images)
```

```
ViewAddress +vViewPort @ \ viewPort
SWAP 2DUP
Ball +asEvenSprite \ the simpleSprite structure
SWAP +EvenImage \ binary data (spriteImage)
ChangeSprite
Ball +asOddSprite \ the other simpleSprite
SWAP +EvenImage \ binary data to be attached
ChangeSprite ;
```

The only difference between this version of "MakeAttached" and the previous version is the addition of "@@" after "DOES>" to indirectly fetch the pointer into CHIP memory where the image resides. This version replaces binary data in the dictionary with a pointer to another pointer to the data in another part of memory. Make sense? Earlier versions of Multi-Forth required a two-step process to turnkey an application and save the blocks of memory pointed to by the IN.HEAP flagged words.

```
TURNKEY" filename" word
OPEN" filename" DUP APPEND.HANDLES CLOSE
```

"Filename" is the name given to the turnkeyed file. "Word" is the Forth word which executes when run. Version 1.21 of Multi-Forth further automates the process, requiring only the first line TURNKEY" filename" word. It takes care of the rest for you.

This technique is not restricted to graphics or sound data. Suppose your program needs a large chunk of memory for some purpose. ALLOTting space increases the dictionary size and makes it more difficult to find enough contiguous space to load the application into memory. If you use GET.MEMORY, you can break the application into smaller pieces, which easily fit into memory's nooks and crannies. This change could mean the difference between the program running or not running (or at least saving you the trouble of deleting another program). Remember, the Amiga is a multitasking machine, and it always pays to be polite.

HOT NEWS AND RUMORS

It's official! There is going to be an ANSI Standard Forth. The X3J14 ANS Forth Technical Committee's first meeting was held in early August in Washington, D.C., with many Forth luminaries in attendance. It will be interesting to see how the committee's efforts progress over the next couple of years.

The Forth Interest Group's RoundTable on GENIE is another item of interest. If you have a question about Forth, GENIE is the place to ask. Everybody who is anybody in Forth appears to have joined (including me). Hope to see you there. Just type "FORTH" at any main menu prompt.

Also, just a reminder about Creative Solutions' forum on CompuServe: I'm still sysoping the Amiga section, and with recent sales of A500s and A2000s, things are really picking

(continued)

up. Type "GO FORTH" at any prompt to get there. If I can't answer your questions, someone from Creative Solutions will try.

Listing One

```
: word| ( delimiter-addr )
scan error" premature delimiter "
pocket bl over count + c! ( blank delimit the string ) ;
: SpriteLine ( -addr1\addr2 )
bl word| dup@ 16 = not
error" must be 16 characters! "
1+ dup 16+ swap ;
: ?SpritePixel ( character\base-value )
DIGIT NOT ERROR" Illegal Sprite Color" ;
: OR_SpritePlanes ( number\address- )
SWAP 2 /MOD ( separate the two bits )
SWAP 16 SCALE ( slide the low-order bit up a word )
OR ( put them back together )
OVER @ 2* ( move the stored value one place left )
OR SWAP ! ( and OR the new bits into place. ) ;
: SimplePlanes ( image\height- )
0 DO SpriteLine
DO IC@ 4 ?SpritePixel OVER OR_SpritePlanes
LOOP 4+
LOOP DROP ;
: SimpleSize ( height-height\#bytes ) DUP 4* 8+ ;
: Sprite ( height- )
SimpleSize CHIP CLEAR or get.memory \ get back a handle
dup 0= error" couldn't get CHIP memory for sprite"
dup CREATE , in.heap \ lay down the handle after the new word
@ 4+ swap SimplePlanes ;
structure AttachedSprite
simpleSprite STRUCT: +asEvenSprite
simpleSprite STRUCT: +asOddSprite
structure end
: OR_AttachedPlanes ( char\even sprite\odd sprite- )
LOCALS| odd even |
DUP 4/ odd OR_SpritePlanes \ shift the two MSB's
3 AND even OR_SpritePlanes ; \ mask the two lowest bits
: AttachedSize ( height-height\offset\total size )
SimpleSize DUP 2* 2+ ; \ two sprites plus a two-byte pointer
```

```
: AttachedPlanes ( image\height\offset- )
LOCALS| offset |
0 DO SpriteLine
DO IC@ 16 ?SpritePixel \ allows characters 0-F
OVER DUP offset + OR_AttachedPlanes
LOOP 4+ \ increment the pointer
LOOP DROP ;
: Attached ( height- )
AttachedSize CHIP CLEAR or get.memory
dup 0= error" couldn't get CHIP memory for attached sprite"
dup CREATE , in.heap \ mark it for turnkey j
\ the hunk address
LOCALS| hunk offset height |
offset 2+ hunk W! \ lay down offset to attached image
SPRITE ATTACHED hunk 2+ offset + ! \ set attach bit
hunk 6+ height offset AttachedPlanes ;
: +EvenImage ( addr1-addr2 ) 2+ ;
: +OddImage ( addr1-addr2 ) DUP W@ + ;
struct AttachedSprite Ball
15 Ball +asEvenSprite +ssHeight W!
15 Ball +asOddSprite +ssHeight W!
structend
: MakeAttached ( name ( height- )
Attached \ CREATE is imbedded here
DOES> ( - )
@@ \ fetch the handle
ViewAddress +vViewPort @ SWAP 2DUP
Ball +asEvenSprite SWAP +EvenImage ChangeSprite
Ball +asOddSprite SWAP +OddImage ChangeSprite ;
15 MakeAttached Ball
0000007777000000
0000754444570000
00A6544334456A00
0086544334456800
0B876544445678B0
0B987665566789B0
ECA9877777789ACE
EDBA99888899ABDE
EEDCBAAAABBCDEE
0FEEDCCCCCDEEF0
0FFEEEEEEEEEF0
00FFEEEEEEFF00
00FFFFFFF00000
0000FFFFFF0000
000000FFFF000000
```

•AC•



"...it is small, installs with just a double-click on an icon...a welcome utility for Amiga users."

—INFO magazine

"...Without a doubt, the most important program this month is GOMF. All Amiga users are sick and tired of losing running programs and data to the dreaded Guru, and anything that can stop this terrible syndrome deserves a round of applause!"

"...So if Gurus are getting to you, add GOMF to your Amiga environment and you can start saying goodbye to the GURU too!"

—Commodore magazine

Now GOMF is even better!
Version 2.0 NOW shipping

NEW FEATURES include:

- Automatic removal of ALL errors (task held, GURU, etc.)
- Preserves low memory auto vectors.
- NUKE those pesky CANCEL requestors once and for ALL!
- Eliminate ANY task or device AT ANY TIME.
- auto-configures for 68000/68010/68020.
- CLI and Workbench compatible.
- Extensive, easy to understand documentation.

only \$39.95

New Products for All AMIGAS from HyperTek/Silicon Springs

Deluxe MIDI interface

Simply the BEST full-featured MIDI interface available for the AMIGA 500/1000. Fully compatible with all programs that use serial MIDI standard output, ONLY the DELUXE MIDI INTERFACE offers you: SERIAL PORT Pass-thru, MIDI IN, MIDI THRU, and two SWITCHABLE THRU/OUTS! This is the optimum configuration for anyone with more than one synth or other MIDI device. No THRU on your synth(s)? No problem! Simply flip a switch on the Deluxe MIDI Interface, and avoid costly THRU boxes! Includes 8 foot cable, 1 year warranty.

Please Specify model 500/1000.....\$89

TTL Hi-res Monitor Adapter

For the Amiga 500/1000/2000. Plugs into RGBI port for ULTRACRISP flicker-free high resolution monochrome video output. Perfect for HI-RES GRAPHICS, WORDPROCESSING, CAD, DESKTOP PUBLISHING, BUSINESS, etc. ANY application in ANY resolution is SHARPER and CLEARER with the TTL Hi-res Monitor Adapter. Perfect for use with the low-cost (under \$100) Commodore 1901 monitor or Monitor 80. Includes easy installation instructions, a disk with a special Hi-Res WorkBench font and RGB port pass-thru (a 1080 RGB monitor may be used simultaneously). 1 year warranty.

BE KIND to your EYES!

(Monitor not included).....\$99.95

Light Pen

Designed to work with ANY Amiga program, the LIGHT PEN and DRIVER allows use of both your mouse and a pro-quality LIGHT PEN for the ultimate in precision graphics. Perfect for PAINTING, DRAWING, freehand SKETCHING, CAD and virtually ALL other AMIGA programs. Software features include: Single-pixel precision, variable sensitivity, ZOOM mode, button toggle, etc. Includes Intell Systems' DT-184A high quality two button lightpen. MAIL ORDER SPECIAL! While supplies last, we'll also include a handy PEN-HOOK that attaches easily to your monitor to hold the AMIGA LIGHTPEN when not in use! (500/1000/2000).....\$129.95

All products are NOW SHIPPING! Please make cheque or money order payable to:

HyperTek/Silicon Springs

#120-1140 Austin Ave. Coquitlam, BC Canada V3K 3P5
Phone (604) 939-8235 Dealer Inquiries Invited.

All orders add \$4 shipping. All prices in US\$
Order by phone! VISA, Mastercard, AMEX welcome.
U.S. orders add \$3 handling fee.

AMIGA is a registered trademark of Commodore-Amiga, Inc.

The Big Picture

by Warren Ring

This month I will show the system calls required to get characters in and out of the CLI, and an example of opening, reading, and closing disk files on the Amiga.

Learning About Assembler on the Amiga

In my quest to learn assembler on the Amiga, I toured my favorite book store, Op Amp Technical Books (1033 N. Sycamore Avenue, Los Angeles, CA 90038 (213) 464-4322). I found 21 different Amiga titles. This book store is by far the best I have seen for computer books. Unfortunately, none of the books dealt with the procedures to perform system calls from assembler programs. Only one discussed assembler language on the Amiga, and it concentrated on 68000 op codes and didn't deal at all with system calls.

While perusing messages passing through some BBSs, I have noticed one book keeps surfacing: *Programmer's Guide to the Amiga* by Robert Peck (Sybex). I have a copy of this book. Although it is written primarily for C users, it has much information that applies to *all* users. It tells *why* you perform certain procedures, as well as how you perform them. Message traffic on the BBSs indicates that many of the C examples don't work. From the little I've read, however, this book looks well worth the investment, even for assembler applications.

I also strongly recommend *68000 Assembly Language Techniques for Building Programs* by Donald Krantz and James Stanley (Addison-Wesley). This book tells you everything you need to know about generic 68000 programming. It reveals which addressing modes are legal for each instruction, and why special instructions and address modes are so great.

Get the Right Tools for the Job

Let's talk about software tools. To write and run assembler programs, you need an editor, an assembler, all the "include" files containing system macros and equates, and a linker. Much of the work on the AmigaDOS was reportedly performed on the Metacomco assembler. That bit of information told me that the assembler itself was probably

well debugged. I tried two assemblers available on Fish disks. One crashed on takeoff; the other would not handle XDEF and ASCII statements properly. I also discovered there were no available include files available. My advice is pay the \$80, and get the Metacomco assembler. This package also has the include files and a linker. Personally, I don't use their editor. I use Uedit from one of the Fish disks. Any Amiga club can get Uedit for you. It's not Word Star, but it's pretty good.

Another tool you need is the *Amiga ROM Kernel Reference Manual: Libraries and Devices*. I consider this strictly a library call reference work. It tells you which values to place in each register for each system call. (I have seen at least one paperback book with similar information.)

Now, you need a couple of batch files. In the directory you have assigned as C:, create a file called "as," and insert the following two lines:

```
.KEY name
assem <name>.asm -o <name>.o -l <name>.lst
```

Next, create a file in the same directory called "lk," and insert the following two lines:

```
.KEY name
alink <name>.o to <name> library llbs:amiga.lib
```

Put the amiga.lib file, which you cannot go anywhere without, into the directory you have assigned to LIBS:. Rename your "execute" file in your C: directory to "do." Give your assembler program filename an ".asm" extension ("test.asm," for example). When you want to assemble it, enter "do as test," a test.o file containing the object code, and a test.lst file containing the listing is created. Assuming it assembles with no errors, enter "do lk test." An executable file called "test" file is very easily created.

Also, copy the "include/exec/types.i" and "include/libraries/dos.i" files from the Metacomco disk to the directory you have assigned as LIBS:.

(continued)

People Meter

the ultimate human interface...



...measures your stress level through innovative hardware. And golly gosh, do we have software! From a full-featured arcade game with sampled sound, to bar and analog graphs for your WorkBench, the five included programs will knock your socks and dry your eyes! On the serious side, the People Meter allows you to fully explore the realm of stress management.

Requires an Amiga 500, 1000, or 2000 with at least 512K and Kickstart 1.2. Available from your Amiga dealer or order direct for \$59.95 plus \$3.50 shipping. CA residents please add 6.5% state tax.

Aminetics P.O. Box 982-205, Whittier CA 90608, (213) 698-6170

Amiga is a registered trademark of Commodore-Amiga, Inc.

A First Test

Every time I get my hands on a new language or computer, I subject the new language/computer combination to a test that really shakes out any gaps. That test consists of writing a file hex dump routine. If you've used the CP/M programs DDT, SID, or ZSID, or the MS-DOS program DUMP, or the HTYPE function of the Amiga program DUIII, you know what a hex dump looks like. A file hex dump program displays a file's data first as a group of hexadecimal numbers, and then as a group of ASCII characters, 16 per line.

To write a file hex dump program, you must open a file, read individual bytes, send characters out to the console, look for a key pressed to stop the display, handle bytes as integers, as well as ASCII characters ... just about every major system function in a language is needed. The manuals must also be written clearly enough for you to pick out the information without much difficulty. Well, the Amiga has proven to be a challenge, but not an insurmountable one.

The program in Listing 1 is not a hex dump program, but rather a "type" program. I have shown a type program here to keep the example short, while retaining all the major

features of a hex dump program. You don't necessarily need a type program, but because you can see how system calls are handled. You can also use derivations of these routines if you are having trouble with system calls.

Some Generalities

First, let's cover some generalities about system calls. The AmigaDOS authors, in a real stroke of genius, knew they could not foresee all future system calls that would be needed. So, they created "libraries" of system calls. These are collections of system calls which can be revised. Each library is designated by an ASCII label. A table of jump instructions to the various routines in that library is located at the top end of each library. When you make a system call to open one of these libraries, AmigaDOS looks in ROM and RAM for the library. If it can't find that library, it looks on disk and loads it from there. Sheer genius, no?

Now, you're probably saying, "Ah, but where is the routine to open a library—in a library, perhaps?" The exec library (which contains some system calls) is always resident, and the location of the jump table in that library is the *only* operating system constant known at link time. You must know the library where your favorite system call resides. Make sure that library is "open" before you make the system call. When you make an operating system call, you must know the location of the library which contains the routine you are calling. You must also know a symbolic offset into that library's jump table. This is not as difficult as it sounds. When you make a system call to open a library, the address of the library's jump table is returned. The symbolic offsets are in the `amiga.lib` file, which you simply include at link time as a library.

As you develop a program and include more and more types of system calls, you usually make sure each library you want to access is opened at initialization time at the beginning of the program. You then make sure it gets closed before you exit. Otherwise, some resources wouldn't be freed up. So how do you know to which library a specified system call belongs? That's easy. Look in the *Amiga ROM Kernel Reference Manual: Libraries and Devices*, which contains a listing of all system calls, along with the libraries in which they can be found.

The CLI Window

OK, now let's talk about the CLI window. To do any work with console I/O in the CLI window, you must first open the "dos" library and locate the already open console input and console output "handles." A handle, in this case, is a data packet which describes the console.

(continued on page 72)

PERFORMANCE!

OR HOW TO OUTFRAN THE 'SUN'.

**IT'S SIMPLE.
JUST CALL THE PERFORMANCE EXPERTS
AT CSA, THE AWARD WINNING,
TECHNOLOGICAL LEADER OF
PERFORMANCE ENHANCEMENT
PRODUCTS FOR THE AMIGA!**

Now you can transform your Amiga 500, 1000, or 2000 into a super-powerful, low-cost workstation by sporting a 68020 full 32-bit processor and optional 68881 co-processor "Piggyback" or CPU board from CSA.

CSA. YOUR PERFORMANCE BOOSTER!

Want to improve performance up to a whopping 200%?

Just run with a 14MHz 68020 alone. Now if you want to cross the finish line with up to a 400% performance increase, then add CSA's 32-bit RAM. For programs requiring a high degree of "number crunching", adding a 20MHz 68881 can reduce your running time from several days down to a matter of minutes!

LEAVING 'EM IN THE DUST.

Independent benchmark tests indicate vastly superior performance to the Macintosh II, VAX 8800, and Sun 3/160 Workstation.

Whether you want to speed up your ray trace program from *hours* to *minutes*, or simply reduce the run time on your spreadsheet from *minutes* to *seconds*, call CSA today.

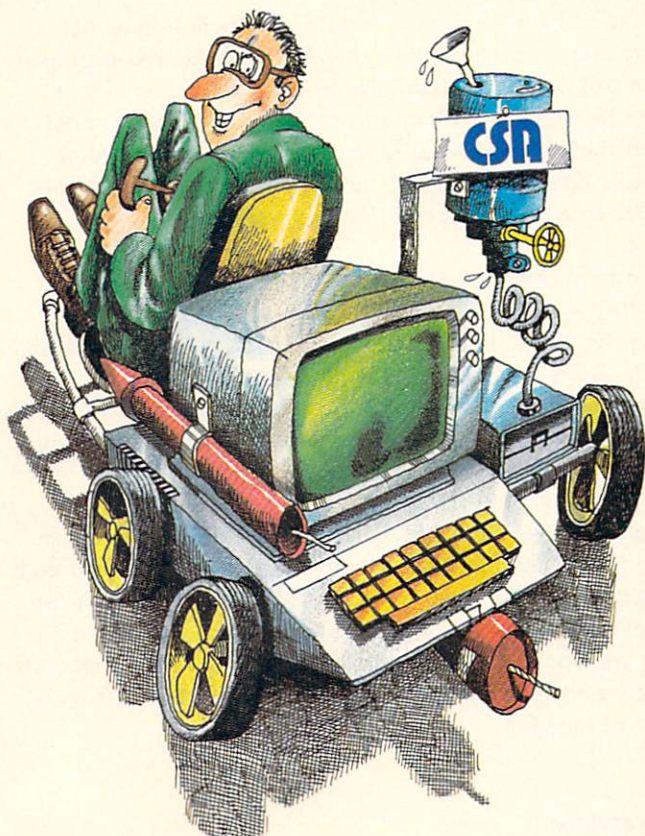
CSA.

We have the best solutions over the Sun.



Computer System Associates, Inc.
7564 Trade Street, San Diego, CA 92121
(619) 566-3911 • TELEX 333-693

Macintosh is the trademark of Apple Computer Corp. VAX is the trademark of Digital Equipment Corp. Amiga is the trademark of Commodore Business Machines, Inc.



DosLibraryHandle is the location of the jump table in the "dos" library. ConsoleInputHandle and ConsoleOutputHandle are the locations of two packets of data used for console input and output, respectively.

To display one character on the CLI console, place that character in the lower byte of D0 and call the DisplayChar routine as shown in Listing 1.

To input a line of characters from a CON: window, call the GetLine routine shown here. When this routine returns, the line of text appears in LineBuffer. Set LineBufLen equal to the longest line of text you expect the user to enter.

```
LineBufLen equ 80

GetLine
;This routine gets a line of
; ASCII characters on the
; console

;Out: LineBuffer = the line of
;      characters

MOVEM.L D0-D3/A0/A1/A5,-(A7) ;Push registers

MOVE.L ConsoleInputHandle,D1 ;Read a line of
MOVE.L #LineBuffer,D2        ; characters
MOVEQ #LineBufLen,D3
MOVE.L DosLibraryHandle,A5
SYS Read(A5)

MOVEM.L (A7)+,D0-D3/A0/A1/A5 ;Pop registers

RTS ;Return

LineBuffer
DS.B LineBufLen ;Line buffer
CNOB 0,2 ;(Align on a word boundary)
```

Before you exit your program, you call the CloseSystem routine.

The routine used for inputting characters from the CLI window is a line input routine. It is not a character input routine. This discrepancy means several things. First, this routine does not return until you press the return key. Second, if you press the backspace key, the line input routine deletes the previously entered character, and backspaces the cursor—without your program getting involved. Third, you can limit the number of characters you want to take from the line, as defined above by LineBufLen. If you enter more characters than LineBufLen specifies, the extra characters are saved and given to your program the next time you ask for successive lines of characters. This process continues until the character stream you entered is exhausted.

Each line-feed is converted to a "carriage-return and line-feed combination" for display upon output. Typically, Amiga text files have no carriage-return characters—only line-feeds—to save a small amount of disk space. (There is

a side effect here. If you try to print a file by copying it to SER: or PAR:, rather than to PRT:, the carriage-return characters will not be inserted in your listing, making it useless.) I have not yet found a way to input a single echoed character from the CLI window.

Retrieving the Command Tail

Assuming your program was invoked from the CLI, you can retrieve the command tail—that is, the command line following the name of your program. When your program starts up (only if it was invoked from the CLI), register A0 contains the address where the command tail is found. Register D0 contains the number of characters in the command tail. For example, if you enter "copy file1 file2" into the CLI, then "file1 file2" is the command tail. The spaces before "file1 file2" are filtered out, and a line-feed character is appended to the string. In this example, D0 contains the value 12. If you wish to use a filename in a command tail to open a file, you must locate the space or line-feed character following the filename and convert it to a null character (zero byte).

The Icing on the Cake

To assemble the program in Listing 1, type it in as file "typ.asm" and enter "do as typ." If you entered it properly, and put the two include files in your LIBS: directory, it should assemble without error. (If the assembler cannot find the include files in LIBS:, it asks you to insert disk "/libs.") Enter "do lk typ." If you put "amiga.lib" in your LIBS: directory, it should link without error.

You now have an executable file called "typ." You may now type the contents of your typ.asm file by entering "typ typ.asm." This program runs a bit slow on my Amiga 500. There is a good reason for this: We are getting data from the disk file, then displaying it one character at a time. If we increase the buffer size from one character to many more, it would run much faster.

I have not described the disk I/O calls in detail, but their use is commented in the listing, and you can see what each register contains. One important disk I/O call I don't use here is the Seek call, which changes the location in the file we are reading. You can position your read/write location relative to your current position, or relative to the front or end of the file.

Overview

The methods of interfacing assembler language programs to AmigaDOS I present in this column are not necessarily the "right" ways to interface. The descriptions are also not necessarily complete. As far as I have been able to deter-

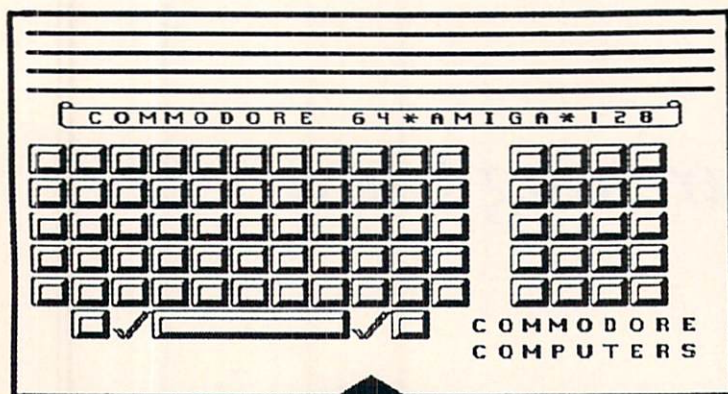
(continued on page 74)

*If You are Searching for a
Monthly
Resource to
the Commodore
AMIGA™*



Amazing
COMPUTING

*Wake up
and Smell the Coffee!!*



617-237-6846

The Memory Location
396 Washington St.
Wellesley, MA 02181
Commodore Specialists

mine, no such "right" ways are documented in any Commodore literature. My methods have been gleaned from a number of sources, primarily methods I have found in the public domain. I would appreciate any insights you may discover about assembler programming on the Amiga. Just send them to me care of *Amazing Computing*, or send a message on BBS "1939" at (818) 368-4248. This BBS has two 2400 baud modems, and has become the *de facto* conferencing BBS for assembler language Amiga hobbyists in the Los Angeles area. Hookups into FidoNet are in the works.

FidoNet is the hobbyist's version of electronic mail, similar to Usenet. If you have a modem and a communications program, such as Comm (available on a Fish disk), you can call the nearest node and send and receive messages with any other hobbyist in the world (if you know his node number), or send and receive messages on any of the conferences the local node monitors. To receive a FidoNet message, you must call your node usually within a few days of the time the message arrives.

There are hundreds of nodes all over the world operated on a volunteer basis by hobbyists. How are the long distance charges covered? Some nodes have night access to WATS lines. Other nodes are covered by interested hobbyists who can afford it or charge a minimal fee, such as \$10 per year.

There are currently five Amiga conferences on FidoNet. Your local Amiga user group should be able to fill you in on the nearest FidoNet node. If not, call me at 1939, and I'll find one for you.

What I Hope to Accomplish in this Column

As time goes by, I would like to accumulate a set of assembler utility routines that simplifies interfacing with AmigaDOS. I would also like a good set of macros, and a set of programming techniques making it nearly as easy to program the Amiga's higher functions in assembler as in C. I've seen it done on other machines, and it can be done on the Amiga. You'll experience the fastest programs you've ever seen on an Amiga, and they'll have capabilities not available in any high-level language.

Next Month

Next month, I'll show how you can put most of this program into a library, so you can use it without ever having to look at it again. We'll cover how to read the function keys and create new text windows. We will also discuss debugging tools. I planned to cover time-sharing an assembler program this month, using one code section and multiple data sections, but that topic will have to wait.

Listing One

```

section code

include "libs:types.i"
include "libs:dos.i"

XREF    _LVOpenLibrary
XREF    _LVOpen
XREF    _LVOClose
XREF    _LVInput
XREF    _LVOutput
XREF    _LVORead
XREF    _LVOWrite
XREF    _LVOCloseLibrary

XREF    _AbsExecBase

CRLF    MACRO
Display 10
ENDM

Display MACRO
JSR     DisplayLine
DC.B    \1,0
CNOP    0,2
ENDM

BufLen equ 128 ;Buffer length

JSR     InitializeSystem

CMPI.L  #1,D0 ;If there was no file
BEQ     NoFile ; specified,
; then jump to NoFile

SUBI.L  #1,D0 ;Convert the terminating
MOVE.B  #0,0(A0,D0) ; line-feed in the command
; tail to a null

MOVE.L  A0,D1 ;Open the file specified in
MOVE.L  #MODE_OLDFILE,D2; the command tail
MOVE.L  DosLibraryHandle,A6
JSR     _LVOpen(A6)
TST.L   D0
BEQ     OpenError ;If the file doesn't exist,
MOVE.L  D0,FileHandle; then jump to OpenError

```



```

Loop
JSR      GetFileChars;Get bytes from the file
TST.B    D0          ;If we have reached EOF,
BEQ      Eof         ; then jump to Eof

JSR      DisplayChars;Display the bytes

BRA      Loop        ;Jump to Loop

Eof
MOVE.L    FileHandle,D1;Close the file
MOVE.L    DosLibraryHandle,A6
JSR      _LVOClose(A6)

Display <'End of file.'>
CRLF

BRA      Quit        ;Jump to Quit

NoFile
Display <'No file was specified.'>
CRLF

BRA      Quit        ;Jump to Quit

OpenError
Display <'There is no such file.'>
CRLF

Quit
JSR      CloseSystem ;(Close the dos library)
RTS      ;Exit the program

;*****
; System calls
;*****

InitializeSystem      ;This routine initializes
                    ; the system

MOVE.M.L D0-D3/A0/A1/A5,-(A7);Push registers

MOVE.L    _AbsExecBase,A6      ;Open the dos library
LEA.L     DosName,A1
MOVEQ     #0,D0
JSR      _LVOPpenLibrary(A6)
MOVE.L    D0,DosLibraryHandle

MOVE.L    DosLibraryHandle,A5 ;Save the console
JSR      _LVOInput(A5)        ; input handle
MOVE.L    D0,ConsoleInputHandle; locally

MOVE.L    DosLibraryHandle,A5 ;Save the console
JSR      _LVOPut(A5)          ; output handle
MOVE.L    D0,ConsoleOutputHandle; locally

MOVE.M.L (A7)+,D0-D3/A0/A1/A5;Pop registers

RTS      ;Return

DosName    DC.B      'dos.library',0
CNOPT      0,2

CloseSystem

MOVE.L    _AbsExecBase,A6      ;Close the dos
MOVE.L    DosLibraryHandle,A1 ; library
JSR      _LVOCloseLibrary(A6)
MOVEQ     #0,D0

RTS      ;Return

GetFileChars
;This routine gets ASCII
; characters from the file

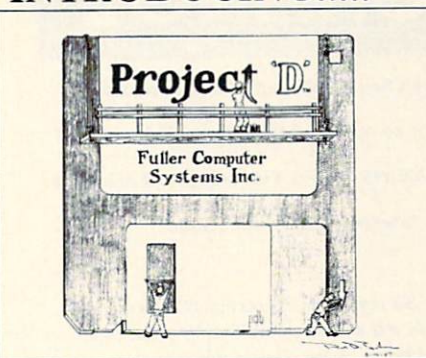
;Out:
; D0.B = the number of
; characters read,
; LineBuffer = the characters

MOVE.M.L D1-D3/A6/A1/A5,-(A7);Push registers

MOVE.L    FileHandle,D1      ;(File specifier)
MOVE.L    #LineBuffer,D2     ;(Buffer location)
MOVE.L    #BufLen,D3         ;(# of chars to get)
MOVE.L    DosLibraryHandle,A5
JSR      _LVORead(A5)
MOVE.L    D0,CharsRead

```

INTRODUCING.....



An
Evolution
in Disk
Utilities
for Amiga™
Personal
Computers!

- F** • An easy to use, friendly and intuitive user interface.
- E** • A powerful and fast disk backup tool that lets you make backups of your copy-protected Amiga software.
- A** • A disk editing tool that lets you edit raw MFM tracks, AmigaDOS sectors and AmigaDOS files (automatically calculating new checksums).
- T** • A disk cataloging tool that lets you maintain lists of your personal, public domain and commercial software.
- U** • A unique backup tool for duplicating other disk formats including MS-DOS/PC-DOS and Atari ST.
- R** • An easy to read, informative user manual is included.
- I** • This product is not copy-protected in any way.
- N**
- G**

NOW SHIPPING!

TO ORDER

\$49.95

EA.

Includes shipping and handling!
Arizona residents add 6.5% sales tax.

Send check or money order to:
Fuller Computer Systems, Inc.
P.O. Box 9222
Mesa, Arizona 85204-0430
Or CALL (602) 835-5018

Amiga is a trademark of Commodore-Amiga, Inc.

Dealer Inquiries Invited

```

MOVE.M.L (A7)+,D1-D3/A0/A1/A5 ;Pop registers

RTS      ;Return

DisplayLine      ;Display Line routine

;This routine displays a line
; of text on the console

;In: The line of text
; immediately follows the
; call to this routine, and
; terminates via a zero byte
; followed by:
; CNOPT 0,2

;Note: This routine expects the
; registers to be pushed
; in a specified order

MOVE.L A2,-(A7) ;(Push registers)
MOVE.L D0,-(A7) ;(Do not optimize this!)
MOVE.L D2,-(A7)

MOVE.L 12(A7),A2;(Set A2 to the address of the
; word following the call to
; this routine)

DisplayLine1
MOVE.W (A2)+,D2 ;If the most-significant byte
MOVE.W D2,D0 ; of that word is a zero byte,
ASR.W #8,D0 ; (Advance A2 by one word)
TST.B D0 ; then jump to DisplayLine9
BEQ.S DisplayLine9
JSR DisplayChar;Display the byte
MOVE.B D2,D0 ;If the least-significant byte
BEQ.S DisplayLine9; of that word is a zero
; byte, then jump to DisplayLine9
JSR DisplayChar;Display the byte
BRA DisplayLine1;Jump to DisplayLine1

```

(continued)

INTERCHANGE

Share objects between Sculpt 3D and VideoScape 3D

Use Sculpt 3D as an editor to create objects for VideoScape 3D

Use Sculpt 3D to make HAM ray-traced Videoscape 3D schemes

Full Intuition interface for all functions

Interchange™ converts Sculpt 3D objects to VideoScape 3D objects and back again. Save hours of work and tedious calculations by using Sculpt 3D to make VideoScape 3D objects. Share objects with others.

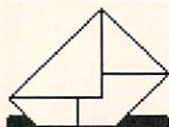
Price \$49.95

Send check or money order only. Please include \$3.00 postage & handling. WI and MA residents add 5% sales tax. This product requires Sculpt 3D and VideoScape. It is not a stand-alone animation program. Send a stamped, self-addressed envelope for a catalog of Syndesis products. Dealer inquiries invited. Interchange is a trademark of Syndesis. Sculpt 3D and VideoScape 3D are registered trademarks of Byte by Byte Corporation and Aegis Development, respectively.

SYNDESIS

20 WEST STREET
WILMINGTON, MASSACHUSETTS 01887
617 • 657 • 5585

MOVING?



SUBSCRIPTION PROBLEMS?

Please don't forget to let us know. If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

```

DisplayLine9
    MOVE.L A2,12(A7):(Set the return address to A2)

    MOVE.L (A7)+,D2:(Pop registers)
    MOVE.L (A7)+,D0:(Do not optimize this!)
    MOVE.L (A7)+,A2

    RTS                                ;Return

DisplayChar                                ;Display Character routine

;This routine displays one
; ASCII character on the
; console

;In: <D0.B> = the character

MOVEM.L D0-D3/A0/A1/A5,-(A7):(Push registers)

MOVE.B D0,DisplayCharx ;Output a character
MOVE.L ConsoleOutputHandle,D1:(File specifier)
MOVE.L #DisplayCharx,D2:(Buffer location)
MOVEQ #1,D3:(# of chars to display)
MOVE.L DosLibraryHandle,A5
JSR _LVCWrite(A5)

MOVEM.L (A7)+,D0-D3/A0/A1/A5:(Pop registers)

    RTS                                ;Return

DisplayChars                                ;Display Characters routine

;This routine displays
; ASCII characters on the
; console

;In: LineBuffer contains the
; characters,
; CharsRead = the number
; of characters in
; LineBuffer

MOVEM.L D0-D3/A0/A1/A5,-(A7):(Push registers)

MOVE.B D0,DisplayCharx ;Output a character
MOVE.L ConsoleOutputHandle,D1:(File specifier)
MOVE.L #LineBuffer,D2:(Buffer location)
MOVE.L CharsRead,D3:(# of chars to display)
MOVE.L DosLibraryHandle,A5
JSR _LVCWrite(A5)

MOVEM.L (A7)+,D0-D3/A0/A1/A5:(Pop registers)

    RTS                                ;Return

LineBuffer
    DS.B BufLen
DisplayCharx
    DS.B 1 ;Character buffer
    CNOP 0,2
CharsRead
    DS.L 1

;*****
; Variables
;*****

    CNOP 0,4
DosLibraryHandle
    DS.L 1
FileHandle
    DS.L 1
ConsoleInputHandle
    DS.L 1
ConsoleOutputHandle
    DS.L 1

end
    
```

•AC•

By John Steiner

Bug Bytes

The Bugs & Upgrades Column

Since starting this column, I have learned many new features of Amiga software ... new to me, at least. One such "new" feature involves Gold Disk's **PageSetter** and **LaserScript**. As a regular **PageSetter** user, I was surprised to learn that it reverses text (white characters on a black background). If you were not aware that you could reverse text, all you need to do is make a box for text, fill it with black, then put \R in front of the text to be placed in the box. It reverses and prints properly (white characters on black) from **PageSetter** or **PagePrint** on a dot-matrix printer.

This point brings up the bug I am reporting here: **LaserScript** does not reverse the text when printing to a Postscript laser printer. An update is available from Gold Disk. Dr. Rod Ludwig also found a solution to the problem which can be used until you get the updated version from Gold Disk. When using **LaserScript**, print the file to disk, rather than to the printer. A Postscript file is not a text file at all. Rather, it is a program, and Postscript is a page description language. **LaserScript** takes a **PageSetter** document and generates a program to drive the Postscript laser printer.

Here is Dr. Ludwig's solution. Use a text editor or word processor to load the Postscript file, and use the editor search function to locate the text you want reversed. For example, you might want to reverse the phrase "Amiga Users Group" in your newsletter. Locate the text in the Postscript file, which will appear in parentheses "(Amiga Users Group)." On the lines

above and below the phrase, insert new lines that contain **SETGRAY** commands as follows:

```
1 SETGRAY
   (Amiga Users Group)
0 SETGRAY
```

Use a communications program to send the file to the laser printer.

Aegis Software's **Draw Plus** has a bug apparent to hard disk users. Starting **Draw Plus** from the CLI does not exhibit the bug, but using the Icon to start the program causes a problem with the plot routine. Normally, **Draw Plus** first plots to a disk file before sending the file to the plotter. The file is written to the hard disk normally, but when it starts to send information to the plotter, it reads the plot file on the disk in DF0: (if there happens to be one there), instead of the newly created plot file on the hard disk.

When trying to load a file in **Deluxe Paint** (if you are using a hard disk and do not have floppy drive df1:), you get a requester that says, "Insert disk df1: in any drive." There is no way to change directories at this point because the only requester options are "Retry" and "Cancel." The Cancel option goes directly back to the program without bringing up a disk requester. If you have a drive DF1:, a requester says, "Insert disk in drive df1:." If you insert any disk in DF1:, the load file requester appears normally.

A temporary solution to the above problem with a single drive system is to use **ASSIGN** from the CLI, and

ASSIGN DF1: to DH0: (or any other logical drive you may have including RAM:). Then simply start **Dpaint** as usual. Thanks to Jacques Chatney for reporting those two bugs.

Absoft Corporation has released seven patch notices for version 1.2 of the **AC/BASIC compiler**. These patches are being sent free of charge to all registered users. The patches and a patch program are sent in printed form; the user must type the programs into a text editor or Amiga BASIC's editor, and then compile. A copy of **AC/BASIC** is then inserted in the drive, and the patch program is executed. The program modifies the copy of **AC/BASIC**, correcting the problem that the patch was designed to fix. If you have had problems with certain **AC/BASIC** commands, you can use this patch list.

Patch	Description
2.01	Fixes INSTR function
2.02	Prevents CVL from eventually crashing Amiga
2.03	Fixes internal housekeeping in DECLARE FUNCTION
2.04A/B	Allows LOF & LOC to work on huge files
2.05A/B	Fixes handling of serial port ("COM1")
2.06	Fixes bug in PRINT USING with commas
2.07	Fixes bug in incremental garbage collection

Absoft is also working on another release of **AC/BASIC**, due out before January 1, 1987. This release should fix the aforementioned bugs, as well as other problems which cannot be patched. This new release will be

(continued)

distributed free to registered users only. Absoft encourages owners of AC/BASIC who have not yet registered their compilers to do so soon.

Robert Woell reports a new release of **Absoft's Fortran compiler**, version 2.3. Mr. Woell beta tested the 2.3 release and noted a few improvements. First of all, he previously used a very early release of 2.2, and there was a problem with data in common blocks that was fixed in later shipments of 2.2.

Some calls to Amiga functions have also been fixed. One major improvement in the new version allows the compiler to use RAM: files as a work area, if you have enough memory or your programs are small. This change sped up compile times by at least a factor of 2 on the tested programs. Extensions to the language have been added, so subroutines can be loaded and unloaded from memory under program control. If you sent your registration card, the upgrade will cost

you \$50; if you didn't send the card, send \$50 *and* your disk.

Send your registration card and technical questions to: Absoft Corporation Technical Support, 2781 Bond Street, Auburn Hills, MI 48057 (313) 853-0050 Telex: 235608

MicroIllusions has announced an update for **The Faery Tale Adventure**. If you cannot enter the building that is due north of the tavern in Tambrey, you need the update. Check the save function by beginning an adventure and entering the save command. If you end up in the CLI after the save, you need the update. To get the update, send in your original disk (and registration card, if you haven't already), to: MicroIllusions, 1748 Chatsworth Street, Granada Hills, CA 91344 Attn: Faery Tale Update

Electronic Arts has upgraded **Arctic Fox** to run correctly under 1.2 with more than 512K of memory. The upgrade costs \$7.50, and EA asks that

you return the entire Arctic Fox package. Send all upgrade requests to: Electronic Arts Amiga Upgrades, Box 7530, San Mateo, CA 94403

Electronic Arts will not accept telephone upgrades. Send check, money order or Visa/MC credit card information, and allow 4-6 weeks delivery.

According to a notice left on People Link, Dan Browning of Progressive Peripherals and Software, has an upgrade to **CLImate**. The newest version of CLImate supports virtual disk VD0:. To upgrade, send your disk to: Progressive Peripherals and Software (Attn: Roy), 464 Kalamath Street, Denver, CO 80204 (303) 825-4144

Thanks to all those who have passed along tips on upgrades and bug solutions. If you have any information you wish to pass along to our readers, send it to: John Steiner/Bug Bytes c/o *Amazing Computing*, P.O. Box 869, Fall River, MA 02722

•AC•

COMPUTER MART

*C2090
A500-HB*

Your Texas Amiga Source
Immediate Access to over 400 Amiga Titles.
Prices too low to print!

We stock Amiga Software and Peripherals
For A500, A1000, & A2000.

Mon. thru Fri. 10:00 AM-7:00 PM, Sat. 12:00-5:00 PM

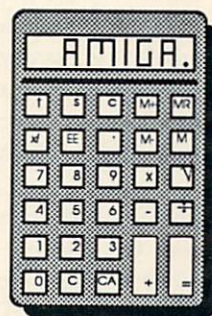
CALL TOLL FREE
800-443-8236

CUSTOMER SERVICE
409-560-2826

Computer Mart • 105 Lynn Street • Nacogdoches, TX 75961

BSpread

by Bryan Catley



A full-featured spreadsheet program written in AmigaBASIC™

BSpread is a full-featured spreadsheet program, written in AmigaBasic, and designed for those of you wondering whether you should spend the money on a commercial product. BSpread lets you become familiar with what a spreadsheet is, what it offers, and what features are best suited to you. It is powerful enough to handle most household requirements, and may even be powerful enough to satisfy the requirements of a small business. Who knows, once you start using BSpread (especially if you compile it with Absoft's AC/Basic compiler), you may not want to purchase that commercial product after all.

Unlike many other Amiga spreadsheets, BSpread makes extensive use of the Amiga's user interface (Intuition) and, accordingly, is very easy to use. Some of its features are: 40 columns by 50 rows, for 2000 cells of a width of 5, 10, 15, or 20 characters; automatic recognition of numeric input which may be displayed with either zero or two decimal places; the first columns (containing row titles) may be held on the screen, while columns to the far right of the sheet are being worked on; columns and rows may be inserted or deleted, causing all other columns or rows to shift as appropriate; a cell's contents may be propagated to the right or down any number of cells; any number of the currently displayed columns may be printed or saved in an ASCII format for export to a word processor; keyboard shortcuts are available for the most commonly used commands; and many others.

Entering BSpread and Checking It Out

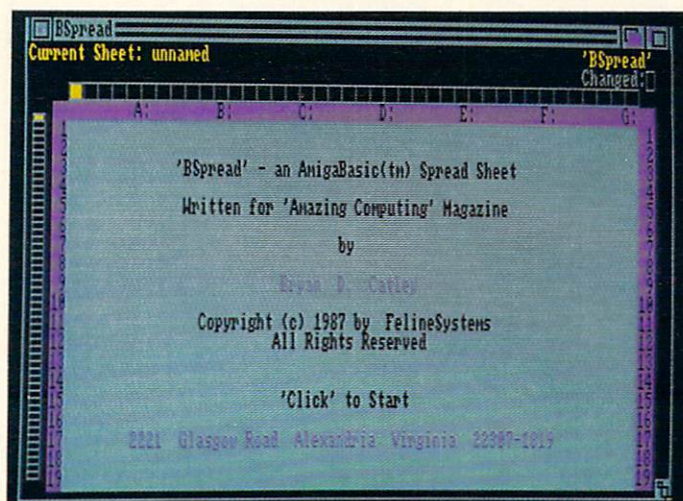
BSpread is a long program, close to 1000 lines. Don't try to enter it all at one sitting, and save your work regularly. There are no special requirements to run BSpread, other

than having at least 512K of memory. It uses the standard AmigaBasic output window (on the Workbench screen), and there are no requirements for special libraries or bmap files. By the way, when entering the program, please leave the titles intact.

Once you have entered BSpread, test it menu item by menu item, until you are reasonably sure you have found all your typos. You will probably not be able to exercise every statement of code, but if you use every menu item, you will come close.

You may also wish to change the "ON ERROR ..." statement

near the beginning of the program to a comment while you are performing this test. If you don't make the change, you'll receive an error message when AmigaBasic finds a typo, without any indication of which statement caused it. Of course, don't forget to remove the "REM" or "" before using the program for any real purpose. If you don't, and an error occurs, you could lose all of your work for that session.



You will also notice "" \$IGNORE ON" and "" \$IGNORE OFF" remarks around a "CLEAR ,25000" statement under the "Xit:" label. These remarks are for the benefit of Absoft's AC/Basic compiler. They may be removed if you do not expect to compile BSpread. In fact, you should remove them because they will slow the exit processing considerably with the interpreter.

Some BSpread Generalities

When you bring up BSpread, you are presented with the working screen which contains the program titles. Clicking

(continued)

the left mouse button removes the titles and places you in the working environment. It also insures the BSpread window is the current one. Notice that rows are labeled on both sides, while the columns are labeled across the top at the right side of each column. Column names range from A to Z, and AA to NN.

Just above the column labels is a row of 40 small boxes, with the left-most one highlighted. Each box represents which column appears as the left-most column on the display. Just click in any box to change the column display. The column of 50 small boxes to the left of the screen performs the same function for the display of rows. As you change columns and rows, notice the delay before you can make another selection. This occurs because the program is redrawing the display area, even if it happens to be blank.

The top line of the screen displays the name of the current spreadsheet; the second line is reserved for error messages, input, and a visual flag which indicates when the spreadsheet has changed from its original condition.

BSpread operates in two modes: Project and Cell. As the names imply, Project Mode gives you control over the entire spreadsheet. Cell Mode allows you to work with individual cells, and specific columns and rows of cells. Cell data and formulas are entered into the sheet in Cell Mode. When you start, you are automatically in Project Mode with only the two left-most menus available. As you move the pointer around the display area, each cell is highlighted; just click in a cell to select it and to move to Cell Mode. Now the right-most two menus are available. You move back to Project Mode by selecting the Exit menu item or pressing the "X" key.

Spreadsheet names, which you can choose, always have ".bsp" appended to them. You may enter this suffix yourself, or let BSpread do it for you. Similarly, ASCII files are always given a ".txt" suffix.

When responding to requests for input, you are presented with the current value (if any), with a cursor over the first character. Pressing the "esc" key clears any current value to null; the cursor left and right keys move the cursor back and forth over the current value; the "back space" key

removes the character to the left of the cursor; the "del" key deletes the character under the cursor. New input is inserted at the current cursor position.

When running BSpread under the interpreter, you must wait a second or two for the cursor to appear the first time you are prompted for input (This is caused by the interpreter rearranging memory to accommodate the sub-program which handles the input. It does not occur again.). The input routine is a variation of my "Basic Input" sub-program which was fully described in *Amazing Computing* Volume 2, Number 5.

The BSpread Menus

BSpread is, for all intents and purposes, menu-driven. Examining the four menus is a good way of learning what BSpread can do for you. Remember, many menu items have keyboard options (shown in the menus themselves and in parentheses below). These optional keyboard selections

are not the familiar Amiga-key combinations. AmigaBasic does not support this option, and to access it via library routines would use too much memory anyway. Thus, an ordinary key press will have to suffice!

The Project Management I Menu is the first of two menus which give you control over the entire spreadsheet. Each option on the menu is described below.

	October		November		Totals
	Income	Expenses	Income	Expenses	
Commissions	2800.00		2800.00		*****
Interest	1500.00		1750.00		*****
Interest	250.00		255.00		*****
Mortgage Payment		900.00		900.00	*****
Car Payment		250.00		250.00	*****
American Express		210.00		150.00	*****
Visa		50.00		75.00	*****
MasterCard		25.00		36.00	*****
Phone Company		16.75		27.75	*****
Electric Company		22.50		35.25	*****
Gas Company		41.60		52.30	*****
Total for Month	*****				*****
YTD Total	*****				*****

Load (L) asks for the name of a previously saved spreadsheet. You may optionally include the ".bsp" suffix. The spreadsheets can be loaded from any directory on any disk, simply by specifying the full path name.

Save (S) saves the current spreadsheet under the current path and name. If there is no current name, you are prompted for one.

Save As (A) requests a new name before saving the current spreadsheet. You may optionally include the ".bsp" suffix. The spreadsheets may be saved on any disk, in any existing directory, simply by specifying the full path name.

Text File (T) saves all 50 rows of a specified number (prompted for) of the currently displayed columns in an

ASCII format, after requesting a file name. You may optionally include the ".txt" suffix. This format is suitable for loading into a word processor.

Print (P) prints all 50 rows of a specified number (prompted for) of the currently displayed columns.

Quit (Q) terminates BSpread and returns to AmigaBasic (or to the system if you are using a compiled version).

The Project Management II Menu is the second of the two menus giving you control over the entire spreadsheet. Each option on the menu is described below.

Calculate (C) resolves all the formulas in the spreadsheet in a left-to-right, top-to-bottom sequence with the results displayed as appropriate.

Go To Cell... (G) requests a cell name, and then repositions the display so the named cell is in the upper left-hand corner of the display area.

Clear Sheet removes *all* entries from the current spreadsheet, resulting in a clean sheet.

Reset Sheet replaces all displays of formula calculations with asterisks.

Cell Size 5,10,15,20 allows you to choose the width of the cells you wish to be working. You may change widths at any time. The current width is indicated with a check mark. If you select a new width which is smaller than the current data, the data will be truncated on the right if it is character data, or replaced with "**LEN*" if it is numeric data. Note that all cells must always be the same width.

Two Decimals states that all numeric values are to be displayed with two decimal places. (Checked when selected.)

No Decimals states that all numeric values are to be displayed as whole numbers. (Checked when selected.)

Hold 1st Cols/Release Cols causes the first n columns (prompted for) to be fixed on the display, or to be released after being fixed. This option toggles back and forth, and allows row names to remain in place, regardless of which columns are in use.

Make a Shell causes all numeric values to be removed from the spreadsheet, and all formulas to be reset to asterisks. This option allows a new spreadsheet "shell" to be created, so you do not have to reenter a standard series of column and row names when you want to create a new sheet in the same format.

THE DIRECTOR™



Professional display and animation language for the Amiga™

Envision a creative freedom you've only dreamed about. Imagine page flipping, color cycling, text generation, even IFF ANIM animations, all combined *at the same time* on the same screen. Now, from the simplest slideshow to the most sophisticated desktop video production, that dream comes true with the Director.

- Use any IFF images, any resolution, any number of colors
- Fades, Dissolves, Blits, Wipe, Stencils
- Page flip full or partial screens
- Preload images, fonts and sounds up to your memory limit
- Flexible script-based structure
- Basic-like vocabulary: For/Next, Gosub/Return, If/Else/Endif
- Arithmetic expressions, random number generator, variables
- Execute AmigaDOS commands from the script
- Text string and file input and output
- Keyboard and mouse interaction
- Digitized soundtrack module
- Supports HAM and overscan
- Supports IFF ANIM playback
- Built-in drawing commands
- No copy protection
- And much more . . .

DEMO DISKS \$10 each
Probe Sequence (512k)
RGB (1 meg)



The Right Answers Group
Department C
Box 3699
Torrance, CA 90510
(213) 325-1311

Amiga is a trademark of Commodore-Amiga, Inc.

\$69.95

Check or money order
payable to: Right Answers
Plus \$3 shipping and handling,
Calif. residents add 6% sales tax

The Cell Management I Menu is the first of two menus which give you control over individual cells. Each option on the menu is described below.

Cell Data (D) allows you to enter new data, or edit existing data, in the currently selected cell. BSpread automatically recognizes character and numeric data, displaying it left or right justified in the cell. Numeric data may be entered with a leading "-" (to indicate a negative value) and/or with a "." (to indicate decimals). Any character data you enter should *not* include a "-" or a "." prefix.

Cell Formula (F) allows you to enter a new formula, or edit an existing one, in the currently selected cell. The entered formula is never displayed on the screen (its length may exceed the cell width.), but rather is shown as asterisks when unresolved, or as the actual value when resolved.

Zero Cell causes the current cell to be reset to zero.

Blank Cell causes the current cell to be reset to blank.

Copy Cell is copied to a specified cell (prompted for). The original cell remains intact.

Move Cell moves the current cell to a specified cell (prompted for). The original cell is reset to blank.

(continued)

Propagate Right causes the current cell to be propagated to the right a specified (prompted for) number of times.

Propagate Down causes the current cell to be propagated down a specified (prompted for) number of times.

Go To Cell... (G) requests a cell name, and then repositions the display, so the named cell is in the upper left-hand corner of the display area.

Exit (X) returns you to Project Mode.

The Cell Management II Menu is the second of the two menus providing you with control over individual cells. Each option on the menu is described below.

Zero Column causes all numeric cells in the current column to be reset to zero.

Zero Row causes all numeric cells in the current row to be reset to zero.

Blank Column causes all character cells in the current column to be reset to blanks.

Blank Row causes all character cells in the current row to be reset to blanks.

Insert Column causes a new column to be inserted as the current column. The existing current column, and all others to its right, are shifted to the right with the resulting loss of column 40.

Insert Row causes a new row to be inserted as the current row. The existing current row, and all others below it, are shifted down with the resultant loss of row 50.

Remove Column causes the current column to be removed from the spreadsheet. All columns to its right are shifted left one column.

Remove Row causes the current row to be removed from the spreadsheet. All rows below it are moved up one row.

Cell Name Specifications

To perform certain functions (in response to a "Go To..." request, and in formulas, for example), you must specify, or name, the cell in question. Cells may be identified by their absolute name or by their relative cell name. The relative cell name is only valid in formulas. Cell names may be entered in either upper or lower case; either is perfectly acceptable.

Absolute Cell Names. There are 40 columns named A to NN, and 50 rows named 1 to 50. To request a specific cell, just combine its column and row names. For example, A1, Z19, CC6, and NN50.

Relative Cell Names. This form of cell specification is reserved for optional use within formulas, and must be enclosed within square "[]" brackets. The desired cell is specified as a plus or minus value to both the current column and the current row, represented by C and R respectively. For example, [C-1,R-1] refers to the cell which is one left and one up from the current cell; [C+5,R-10] refers to the cell five positions to the right, and 10 up from the current cell. Both column and row values *must* be specified, even if there is no change (Use a zero in this case—[C+0,R+1], for example.). Column and row values must be separated by a comma.

A relative cell name is obviously longer than an absolute cell name; it often only needs to be entered once, though, and then propagated when a similar formula is required in a number of adjacent cells. Further, relative cell names are usually still valid after new columns or rows have been inserted or deleted! One drawback to using relative cell names is that they are not calculated as quickly as absolute cell names (even with a compiled version of the program).

BSpread Formulas

The heart of any spreadsheet is its ability to process formulas and display the results. In BSpread, formulas may be entered into any empty cell by selecting the "Cell Formula" menu option (or by pressing the "F" key) when in Cell Mode. When the formula is first entered, the cell appears filled with asterisks. It remains this way until the "Calculate" menu item is selected (or the "C" key is pressed) in Project Mode. The asterisks are then replaced with the result of the evaluated formula.

A formula is a series of absolute cell names and/or relative cell references and/or numeric literals separated by arithmetic operators. The numeric literals may optionally start with a "-" sign and may contain a decimal point. The supported arithmetic operators are + (addition), - (subtraction), * (multiplication), and / (division). Parentheses are *not* supported, and operations proceed on a strictly left to right basis.

For example, to find the average of the contents of three cells, you would use a formula similar to: B10+[C+2,R-3]+D10/3. To express N20 as a percentage of NN10, you could use: N20/NN10*100. To add 25 percent to the contents of the cell directly above, you might use: [C+0,R-1]*.25+[C+0,R-1].

Formulas may also reference cells containing other formulas which have already been evaluated. Evaluation proceeds in a left-to-right, top-to-bottom sequence.

In addition to defining your own formulas, you may also use two special, predefined formulas to perform two of the most common calculations quickly and easily. "TOTCOL" totals all numeric cells that are above, and in the same column; "TOTROW" totals all numeric cells to the left, and in the same row.

If a formula references a cell which does not contain a numeric value, an error message is displayed. The asterisks remain in the cell containing the formula which made the invalid reference.

Error Messages

A number of error messages may be generated. The meanings are usually fairly obvious in relation to the situation (Most are concerned with invalid formulas.).

You may also receive an "AmigaBasic Error: n" message, where n represents the AmigaBasic error number you can look up in the AmigaBasic reference manual. At the same time, you are automatically returned to Project Mode where you may recover from the error or save the current sheet before quitting.

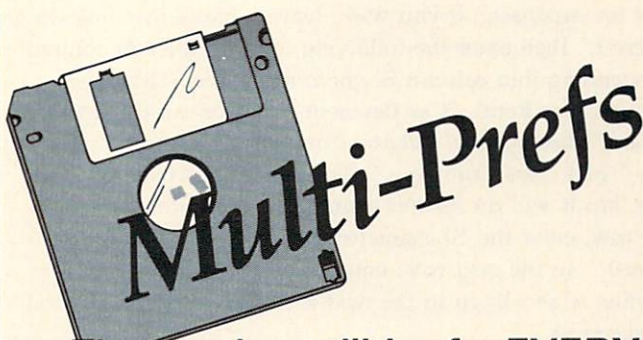
Sometimes you must identify which statement caused the error. To do this, you must recreate the situation, using the interpreter, with the "ON ERROR ..." statement changed to a comment (just as you did during the initial checkout phase). The error is probably just a typo which survived your initial testing!

A Simple Example

If you've never used a spreadsheet, you should look at a simple example. One popular spreadsheet application is household budget management. So, to start things off, let's do a very simple household budget!

First, we must decide what size cell to use, and whether to select zero or two decimals. For this example, the default width of 10 is probably adequate. As far as the decimals places go, it's up to you! Do you round things off to the nearest dollar, or do you like to see exact amounts?

Having made those basic decisions, we now must decide on format. For this example, we'll use two columns for each month, one for income and one for expenses. This means the first two rows of our spreadsheet are reserved for column titles: month names in the first row, and "Income" and "Expenses" in the second. In turn, the names for the



Time saving utilities for EVERY Amiga owner!

- Create multiple preference settings.
- Quickly and easily restore them as needed.
- Edit, Sort, Delete and Undelete any of them.
- Fast, Fun and Easy to use, from WorkBench or CLI.
- Runs on ALL Amigas . . . 500, 1000 and 2000 (V1.2 required).
- Multi-tasking.
- Not copy protected.
- Plus . . . 3 other time saving utilities are included.

Available NOW!!!

Just send \$29.95 plus \$2.00 S&H to:

SOFT-LINK inc. P.O. Box 304, Coventry, RI 02816

Dealer inquiries invited.

Amiga is a registered trademark of Commodore-Amiga, Inc.

individual line items appear on the left side of the spreadsheet. Because more meaningful names require more than 10 characters, we'll use the first two columns for item names.

Okay, load BSpread, click the left mouse button to start, and let's get going!

We'll start with the income line item names. Move the pointer (the cursor will follow) to cell A3 and click. Cell A3 becomes the current cell, and you are now in Cell Mode. Now select "Cell Data" from the "Cell Mgmt I" menu (or press the "D" key). You are prompted for the data to be entered into the cell. Type "Salaries" (without the quotes) and press return. Lo and behold, the word "Salaries" appears in cell A3! Now click in cell A4 and select "Cell Data" again. This time enter "Commissions." Note that you are not be allowed to enter the final "s" because it's the eleventh character (only ten characters are allowed in each cell). Press return, move the pointer to cell B4 and click. Select "Cell Data," enter the "s," and press return. "Commissions" now occupies cells A4 and B4. Using the same technique, enter "Interest" in cell A5. That's it for the income line!

(continued)

Now for expenses. If you wish, leave a blank row following "Interest," then enter the following expense titles in column A, extending into column B where necessary: "Mortgage Payment" (or Rent), "Car Payment," "American Express," "Visa," "MasterCard," "Phone Company," "Electric Company," and "Gas Company." The list is obviously incomplete, but it will do for our example. In column C of the next row, enter the 10 character string "-----" (ten dashes). In the next row, enter "Total for Month" in columns A and B; go to the next row and enter "YTD Total" in column A.

Jumping up to cell C2, enter the first column title "bbbbIncome" (bbbb represents four blanks and causes the title to appear right justified in the cell). In cell D2, enter "bbExpenses." We just need to enter the name of the month. This entry can go in cell C1, D1, or split in the middle between the two. I'd opt for the latter; it looks much neater. Repeat the column heading for a second month in columns E and F. Move over to column G and enter "bbbbTotals" in row 2. (We're only going to do two months in this example.)

There are only two more things to do before we start entering actual data. First, click in the cell which contains the dashes, select the "Propagate Right" menu item, and enter "3" at the prompt. The cell of dashes is now spread across the spreadsheet an additional five columns. Second, enter the formulas. Entering these now, instead of waiting for the data in the spreadsheet to be completed, allows you to monitor the data as being entered. The most obvious formulas for use initially are the predefined "TOTCOL" and "TOTROW" formulas.

Click in the cell in column C immediately below the cell of dashes. Select "Cell Formula" from the "Cell Mgmt I" menu (or press the "F" key) and, when requested, enter "TOTCOL" (without the quotes). The cell is then filled with asterisks, indicating it contains an unresolved formula. While the cell is still current, propagate (or copy) it to the right five columns (as you did with the dashes).

Now click in cell G3 (to make it the current cell), select the "Cell Formula" menu item, and enter "TOTROW" when requested. As before, asterisks appear in the cell. Now use the "Propagate Down" menu item to copy the formula down the column for each line item in your spreadsheet (except for the rows already containing "TOTCOL" formulas, of course).

The only remaining formulas are the Year-To-Date ones. Well, in the first two data columns (C and D), this total is nothing more than a repeat of the months totals from directly above. In the YTD row of column C, enter the formula [C+0,R-1], and propagate it one column to the right. Obviously, this isn't really a formula; it simply copies the

above total. The big advantage of doing this is, if we choose to insert more rows at a later time, this formula is still valid!

For the third data column (and all subsequent columns), we want to add the contents of the cell immediately above the YTD cell to the contents of the cell two columns immediately to the left. This formula looks like: [C-2,R-1]+[C+0,R-1]. Note the use of relative cell specifications. The same formula is now valid for every column! Enter it in the YTD row of column E and propagate it to the right the appropriate number of times (With our simple example, this move is only one column.).

You now have a spreadsheet shell you can save for later use, or which you can start filling in right away (You can always use the "Make Shell" menu item, if you wish to start a similar spreadsheet from scratch.).

Now, enter some data (Click in the desired cell, select the "Cell Data" menu item—or press the "D" key—and enter the data you desire.). Numeric data is recognized and right justified in the cell. Once you have a reasonable amount of data entered, return to "Project Mgmt" mode by selecting "Exit." Save your spreadsheet, and select the "Calculate" menu item (or press the "C" key).

Calculation takes a little time; the entire sheet must be processed, and relative cell names are not calculated as quickly as absolute names. As results are evaluated, the asterisks are replaced with the appropriate values.

That's it! Your first spreadsheet is complete. Experiment with this example before creating your own spreadsheet.

Finally...

As I said earlier, BSpread is a long program. You may want to purchase the Amicus disk which contains it, instead of typing it yourself. If it has not found its way onto an Amicus disk by the time you read this, it shouldn't be too long before it becomes available, so write for it anyway!

Listing One

```

\ "BSpread" - an AmigaBasic™ Spread Sheet;
\ Written for 'Amazing Computing' Magazine
\ by Bryan D. Catley, August 1987
\
\
\ Copyright (c) 1987 by FelineSystems.
\ All rights reserved.
\
CLEAR ,25000: CLEAR ,105000: OPTION BASE 1
MaxCol=40:MaxRow=50:CWidth=10:NumCols=7:Yes=0
Blk=0:Gra=1:Mag=2:Yel=3:m0=0:menuID=0:menuIT=0
n=0:x=0:y=0:z=0:rowtop=0:colleft=0:oldtop=0:oldleft=0
StartCol=1:EndCol=7:ColLite=0:FirstCol=40>LastCol=600
StartRow=1:EndRow=19:RowLite=0:FirstRow=32>LastRow=184
ErrSw=0:RelSw=0:Total=0:CellMgmt=0:Holding=0:HoldCols=0
TempCol=0:TempRow=0:NoGdgt=0:AllOK=-1:Saved=-1
x0=0:row0=0:col0=0:PrintCol0=6:PrintRow0=5
CellValue$="":TypeFlag$="":Response$="":Cell$=""
TempCol$="":TempRow$="":Prompt$="":Prompt2$=""

```



```

a$="":b$="":x$="":y$="":z$="":FileNm$="":TxtNm$="":
Formula$="":ErrMsg$="":op$="":ops$="":!/"
Mask$="#####.##-":Key$="
DIM IPcursor(20),CellLite(165)
DIM ColHdg$(MaxCol),BSpread$(MaxCol,MaxRow,2),MKeys$(2)
GOSUB Initialize:ON ERROR GOTO BasicError
WHILE MOUSE(0)=0:WEND:WHILE MOUSE(0)<>0:WEND
LINE(40,32)-STEP(560,152),Gra,bf

WaitForUser:
IF ErrSw=0 THEN
  COLOR Mag,Blk:LOCATE 2,1
  PRINT"Use Project Menus or Select a Cell.":SPACES(33):
END IF
m0=0:menuID=0:menuIT=0:Key$="":ErrSw=0
MKeys$(1)="LSATPQ":MKeys$(2)="CG"
WHILE m0=0 AND menuID=0 AND Key$=""
  x=MOUSE(1):y=MOUSE(2)
  IF x>FirstCol AND x<LastCol AND y>FirstRow AND y<LastRow THEN
    GOSUB MouseOK
  ELSE
    IF oldtop<>0 THEN PUT(oldleft,oldtop),CellLite%
    oldtop=0:rowtop=0:oldleft=0:colleft=0
  END IF
  m0=MOUSE(0):menuID=MENU(0):menuIT=MENU(1):Key$=INKEY$
WEND
IF Key$<>"" THEN
  FOR n=1 TO 2
    x=INSTR(MKeys$(n),UCASE$(Key$))
    IF x<>0 THEN menuID=n:menuIT=x:n=2
  NEXT
END IF
IF menuID<>0 THEN ON menuID GOTO ProjMgmtI,ProjMgmtII
x=MOUSE(1):y=MOUSE(2):WHILE MOUSE(0)<>0:WEND
IF x>40 AND x<600 AND y>16 AND y<24 THEN NewColumn
IF x>4 AND x<16 AND y>29 AND y<179 THEN NewRow
IF x<FirstCol OR x>600 OR y<FirstRow OR y>184 THEN WaitForUser

CellSelected:
CellCol=INT((x-FirstCol)/(CWidth*8)+StartCol)
CellRow=INT((y-FirstRow)/8+StartRow)
MENU 1,0,0:MENU 2,0,0:MENU 3,0,1:MENU 4,0,1
CellMgmt=1
NewCellSelection:
COLOR Mag,Blk:LOCATE 2,1
PRINT"Use Cell Menus or Click in Another Cell.":SPACES(28):
m0=0:menuID=0:menuIT=0:Key$="":MKeys$(1)="DF$$$$$GX":MKeys$(2)="
WHILE m0=0 AND menuID=0 AND Key$=""
  m0=MOUSE(0):menuID=MENU(0):menuIT=MENU(1):Key$=INKEY$
WEND
IF Key$<>"" THEN
  x=INSTR(MKeys$(1),UCASE$(Key$))
  IF x<>0 THEN menuID=3:menuIT=x
END IF
IF menuID<>0 THEN
  ON menuID-2 GOTO CellMgmtI,CellMgmtII
ELSE
  x=MOUSE(1):y=MOUSE(2):WHILE MOUSE(0)<>0:WEND
  IF x>40 AND x<600 AND y>16 AND y<24 THEN NewColumn
  IF x>4 AND x<16 AND y>29 AND y<179 THEN NewRow
  IF x>FirstCol AND x<LastCol AND y>FirstRow AND y<LastRow THEN
    GOSUB MouseOK:GOTO CellSelected
  ELSE
    GOTO NewCellSelection
  END IF
END IF

NewColumn:
StartCol=INT((x-40)/14)+1:LastCol=600
IF StartCol+NumCols>MaxCol THEN
  LastCol=(MaxCol+1-StartCol)*(CWidth*8))+40
END IF
GOSUB DoColHdg:GOSUB ShoData
IF CellMgmt THEN CellSelected ELSE WaitForUser

NewRow:
StartRow=INT((y-29)/3)+1:LastRow=184
IF StartRow+19>MaxRow THEN LastRow=((MaxRow+1-StartRow)*8)+32
GOSUB DoRowHdg:GOSUB ShoData:
IF CellMgmt THEN CellSelected ELSE WaitForUser

CellMgmtI:
ON menuIT GOTO CellData,CellFormula,ZeroCell,BlankCell,CopyCell
ON menuIT-5 GOTO MoveCell,PropRight,PropDown,CMGoTo,CellExit

CellData:
CellValue$=BSpread$(CellCol,CellRow,1)
TypeFlag$=BSpread$(CellCol,CellRow,2)
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"New Cell Value? ":
x=CWidth:GetData CellValue$,"CHAR",x%
IF CellValue$<>"" THEN
  BSpread$(CellCol,CellRow,1)=CellValue$
  BSpread$(CellCol,CellRow,2)=TypeFlag$
ELSE

```

```

  BSpread$(CellCol,CellRow,1)=""
  BSpread$(CellCol,CellRow,2)=""
END IF
GOSUB PrintCell:GOTO CellChanged

CellFormula:
CellValue$=MID$(BSpread$(CellCol,CellRow,2),2)
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"New Formula? ":
GetData CellValue$,"CHAR",53:CellValue$=UCASE$(CellValue$)
IF CellValue$<>"" THEN
  BSpread$(CellCol,CellRow,2)="F"+CellValue$
  BSpread$(CellCol,CellRow,1)=STRINGS(CWidth,"")
ELSE
  BSpread$(CellCol,CellRow,1)=""
  BSpread$(CellCol,CellRow,2)=""
END IF
GOSUB PrintCell:GOTO CellChanged

ZeroCell:
BSpread$(CellCol,CellRow,1)=""
BSpread$(CellCol,CellRow,2)=""
GOSUB PrintCell:GOTO CellChanged

BlankCell:
BSpread$(CellCol,CellRow,1)=""
BSpread$(CellCol,CellRow,2)=""
GOSUB PrintCell:GOTO CellChanged

CopyCell:
MoveCell:
Prompt$="Target Cell? ":GOSUB GetCellNum
BSpread$(TempCol,TempRow,1)=BSpread$(CellCol,CellRow,1)
BSpread$(TempCol,TempRow,2)=BSpread$(CellCol,CellRow,2)
IF menuIT=6 THEN
  BSpread$(CellCol,CellRow,1)=""
  BSpread$(CellCol,CellRow,2)=""
END IF
IF TempCol>StartCol AND TempCol<=EndCol THEN
  IF TempRow>StartRow AND TempRow<=EndRow THEN
    IF menuIT=6 THEN GOSUB PrintCell
    x=CellCol:y=CellRow:CellCol=TempCol:CellRow=TempRow
    GOSUB PrintCell:CellCol=x:CellRow=y
  END IF
END IF
GOTO CellChanged

PropRight:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"Number of Columns? ":
Response$="":GetData Response$,"INT",2:y=VAL(Response$)
IF Response$<>"" THEN
  IF CellCol+y>MaxCol THEN y=MaxCol-CellCol
  FOR x=CellCol+1 TO CellCol+y
    BSpread$(x,CellRow,1)=BSpread$(CellCol,CellRow,1)
    BSpread$(x,CellRow,2)=BSpread$(CellCol,CellRow,2)
  NEXT
  y=y+EndCol:IF y>EndCol THEN y=EndCol
  x=CellCol:FOR CellCol=x+1 TO y:GOSUB PrintCell:NEXT
  CellCol=x
END IF
GOTO CellChanged

PropDown:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"Number of Rows? ":
Response$="":GetData Response$,"INT",2:y=VAL(Response$)
IF Response$<>"" THEN
  IF CellRow+y>MaxRow THEN y=MaxRow-CellRow
  FOR x=CellRow+1 TO CellRow+y
    BSpread$(CellCol,x,1)=BSpread$(CellCol,CellRow,1)
    BSpread$(CellCol,x,2)=BSpread$(CellCol,CellRow,2)
  NEXT
  y=CellRow+y:IF y>EndRow THEN y=EndRow
  x=CellRow:FOR CellRow=x+1 TO y:GOSUB PrintCell:NEXT
  CellRow=x
END IF
GOTO CellChanged

CMGoTo:
GOSUB GoToCell:GOTO NewCellSelection

CellMgmtII:
ON menuIT GOTO ZeroCol,ZeroRow,BlankCol,BlankRow,InsertCol
ON menuIT-5 GOTO InsertRow,RemoveCol,RemoveRow

ZeroCol:
BlankCol:
COLOR Mag,Blk:LOCATE 2,1:PRINT SPACES(68)
IF menuIT=1 THEN Prompt$=" Ok to zero":Prompt2$=" this column?"
IF menuIT=1 THEN a$="N":b$="0"
IF menuIT=3 THEN Prompt$=" Ok to blank":Prompt2$=" this column?"
IF menuIT=3 THEN a$="C":b$=""
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
IF menuIT=1 THEN LOCATE 2,1:PRINT"Zeroing column..."
IF menuIT=3 THEN LOCATE 2,1:PRINT"Blanking column..."

```

(continued)


```

FOR TempRow=1 TO MaxRow
  IF BSread$(CellCol,TempRow,2)=a$ THEN
    BSread$(CellCol,TempRow,1)=b$
    x=CellRow:CellRow=TempRow:GOSUB PrintCell:CellRow=x
    IF menuIT=3 THEN BSread$(CellCol,TempRow,2)=""
  END IF
NEXT
GOTO CellChanged

ZeroRow:
BlankRow:
COLOR Mag,Blk:LOCATE 2,1:PRINT SPACES(68)
IF menuIT=2 THEN Prompt$=" Ok to zero":Prompt2$=" this row?"
IF menuIT=2 THEN a$="N":b$="0"
IF menuIT=4 THEN Prompt$=" Ok to blank":Prompt2$=" this row?"
IF menuIT=4 THEN a$="C":b$=""
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
IF menuIT=2 THEN LOCATE 2,1:PRINT"Zeroing column..."
IF menuIT=4 THEN LOCATE 2,1:PRINT"Blanking column..."
FOR TempCol=1 TO MaxCol
  IF BSread$(TempCol,CellRow,2)=a$ THEN
    BSread$(TempCol,CellRow,1)=b$
    x=CellCol:CellCol=TempCol:GOSUB PrintCell:CellCol=x
    IF menuIT=4 THEN BSread$(TempCol,CellRow,2)=""
  END IF
NEXT
GOTO CellChanged

InsertCol:
COLOR Mag,Blk:LOCATE 2,1:PRINT SPACES(68)
Prompt$=" Ok to loose":Prompt2$=" column NN?"
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
COLOR Gra,Blk:LOCATE 2,1:PRINT"Inserting column..."
IF CellCol<MaxCol THEN
  FOR TempCol=MaxCol-1 TO CellCol STEP -1
    FOR TempRow=1 TO MaxRow
      BSread$(TempCol+1,TempRow,1)=BSread$(TempCol,TempRow,1)
      BSread$(TempCol+1,TempRow,2)=BSread$(TempCol,TempRow,2)
    NEXT
  NEXT
END IF
FOR TempRow=1 TO MaxRow
  BSread$(CellCol,TempRow,1)=""
  BSread$(CellCol,TempRow,2)=""
NEXT
GOSUB ShoData:GOTO CellChanged

InsertRow:
COLOR Mag,Blk:LOCATE 2,1:PRINT SPACES(68)
Prompt$=" Ok to loose":Prompt2$=" row 50?"
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
COLOR Gra,Blk:LOCATE 2,1:PRINT"Inserting row..."
IF CellRow<MaxRow THEN
  FOR TempCol=1 TO MaxCol
    FOR TempRow=MaxRow-1 TO CellRow STEP -1
      BSread$(TempCol,TempRow+1,1)=BSread$(TempCol,TempRow,1)
      BSread$(TempCol,TempRow+1,2)=BSread$(TempCol,TempRow,2)
    NEXT
  NEXT
END IF
FOR TempCol=1 TO MaxCol
  BSread$(TempCol,CellRow,1)=""
  BSread$(TempCol,CellRow,2)=""
NEXT
GOSUB ShoData:GOTO CellChanged

RemoveCol:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
Prompt$=" OK to delete":Prompt2$="current column?"
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
LOCATE 2,1:PRINT"Deleting column..."
IF CellCol<MaxCol THEN
  FOR TempCol=CellCol TO MaxCol-1
    FOR TempRow=1 TO MaxRow
      BSread$(TempCol,TempRow,1)=BSread$(TempCol+1,TempRow,1)
      BSread$(TempCol,TempRow,2)=BSread$(TempCol+1,TempRow,2)
    NEXT
  NEXT
END IF
FOR TempRow=1 TO MaxRow
  BSread$(MaxCol,TempRow,1)=""
  BSread$(MaxCol,TempRow,2)=""
NEXT
GOSUB ShoData:GOTO CellChanged

RemoveRow:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
Prompt$=" OK to delete":Prompt2$=" current row?"
GOSUB YesNo:IF NOT Yes THEN NewCellSelection
LOCATE 2,1:PRINT"Deleting row..."
IF CellRow<MaxRow THEN
  FOR TempCol=1 TO MaxCol
    FOR TempRow=CellRow TO MaxRow-1
      BSread$(TempCol,TempRow,1)=BSread$(TempCol,TempRow+1,1)
      BSread$(TempCol,TempRow,2)=BSread$(TempCol,TempRow+1,2)
    NEXT
  NEXT
END IF
END IF

```

```

FOR TempCol=1 TO MaxCol
  BSread$(TempCol,MaxRow,1)=""
  BSread$(TempCol,MaxRow,2)=""
NEXT
GOSUB ShoData:GOTO CellChanged

CellChanged:
Saved=0:LINE(609,9)-STEP(6,6),Mag,bf
GOTO NewCellSelection

CellExit:
IF NOT Holding THEN MENU 1,0,1
MENU 2,0,1:MENU 3,0,0:MENU 4,0,0
CellMgmt=0:GOTO WaitForUser

ProjMgmtI:
ON menuIT GOTO LodSheet,SvSheet,SvSheetAs,SvTxt,prStart,Xit

LodSheet:
IF NOT Saved THEN
  Prompt$="Sheet not saved":Prompt2$=" Continue?"
  GOSUB YesNo:IF NOT Yes THEN WaitForUser
END IF
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"Load sheet? ":
LSget:
COLOR Gra,Blk:LOCATE 2,13:GetData FileNm$,"CHAR",39
IF FileNm$="" THEN WaitForUser
IF RIGHTS$(FileNm$,4)<>".bsp" THEN FileNm$=FileNm$+".bsp"
OPEN FileNm$ FOR INPUT AS #1
AfterOpen:
IF ErrSw=1 THEN ErrSw=0:GOTO LSget
LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"Loading sheet ":FileNm$
INPUT#1,CWidth,NumCols,StartCol,EndCol
INPUT#1,LastCol,StartRow,EndRow,LastRow
INPUT#1,Mask$
FOR TempCol=1 TO MaxCol
  FOR TempRow=1 TO MaxRow
    INPUT#1,BSread$(TempCol,TempRow,1)
    INPUT#1,BSread$(TempCol,TempRow,2)
  NEXT
NEXT
CLOSE #1:GOSUB DoColHdg:GOSUB DoRowHdg
MENU 2,5,1:MENU 2,6,1:MENU 2,7,1:MENU 2,8,1
x=CWidth/5+4:MENU 2,x,2
Saved=-1:LINE(609,9)-STEP(6,6),Blk,bf:COLOR Yel,Blk
LOCATE 1,16:PRINT RIGHTS$(FileNm$,52):SPACES(52-LEN(FileNm$))
IF MID$(Mask$,17,1)="" THEN TwoDec ELSE NoDec

SvSheet:
SvSheetAs:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68)
IF FileNm$="" OR menuIT=3 THEN
  LOCATE 2,1:PRINT"Save as sheet? ":
  GetData FileNm$,"CHAR",50
END IF
IF FileNm$="" THEN WaitForUser
IF RIGHTS$(FileNm$,4)<>".bsp" THEN FileNm$=FileNm$+".bsp"
LOCATE 2,1:PRINT SPACES(68)
LOCATE 2,1:PRINT"Saving sheet ":FileNm$
OPEN FileNm$ FOR OUTPUT AS #1
WRITE#1,CWidth,NumCols,StartCol,EndCol
WRITE#1,LastCol,StartRow,EndRow,LastRow
WRITE#1,Mask$
FOR TempCol=1 TO MaxCol
  FOR TempRow=1 TO MaxRow
    WRITE#1,BSread$(TempCol,TempRow,1)
    WRITE#1,BSread$(TempCol,TempRow,2)
  NEXT
NEXT
CLOSE #1:Saved=-1:COLOR Yel,Blk
LOCATE 1,16:PRINT RIGHTS$(FileNm$,52):SPACES(52-LEN(FileNm$))
GOTO WaitForUser

SvTxt:
prStart:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACES(68):z$=""
LOCATE 2,1:PRINT"Number of columns? ":z$=GetData z$,"INT",1
z=VAL(z$):LOCATE 2,1:PRINT SPACES(68)
IF z>NumCols OR z<1 THEN z=NumCols
IF menuIT=4 THEN
  LOCATE 2,1:PRINT"Save as text file? ":
  GetData TxtNm$,"CHAR",48
  IF TxtNm$="" THEN WaitForUser
  IF RIGHTS$(TxtNm$,4)<>".txt" THEN TxtNm$=TxtNm$+".txt"
  LOCATE 2,1:PRINT SPACES(68)
  LOCATE 2,1:PRINT"Saving text file ":TxtNm$
  OPEN TxtNm$ FOR OUTPUT AS #1
ELSE
  LOCATE 2,1:PRINT"Printing sheet..."
END IF
FOR y=1 TO MaxRow
  FOR x=StartCol TO StartCol+z-1
    x$=BSread$(x,y,1)
    y$=LEFT$(BSread$(x,y,2),1)
    IF (y$="N" OR y$="F") AND LEN(x$)>CWidth THEN

```



```

x$="*LEN*"+STRING$(CWidth-5,"*")
END IF
n=CWidth-LEN(x$):IF n<0 THEN n=0
IF y$="C" OR y$="*" OR LEFT$(x$,1)="/" THEN
  IF menuIT=4 THEN
    PRINT#1,x$:SPACE$(n):
  ELSE
    LPRINT x$+SPACE$(n):
  END IF
ELSE
  IF menuIT=4 THEN
    PRINT#1,USING RIGHT$(Mask$,CWidth):VAL(x$):
  ELSE
    LPRINT USING RIGHT$(Mask$,CWidth):VAL(x$):
  END IF
END IF
NEXT
IF menuIT=4 THEN PRINT#1," " ELSE LPRINT " "
NEXT
IF menuIT=4 THEN CLOSE#1
GOTO WaitForUser

ProjMgmtII:
ON menuIT GOTO Calculate,PRGoTo,ClrSheet,ResetSheet,CSz5
ON menuIT-5 GOTO CSz10,CSz15,CSz20,TwoDec,NoDec,HoldRlse
ON menuIT-11 GOTO Template

Calculate:
COLOR Gra,Blk:LOCATE 2,1:PRINT"Calculating...":SPACE$(54):
FOR CellRow=1 TO MaxRow
  FOR CellCol=1 TO MaxCol
    IF LEFT$(BSpread$(CellCol,CellRow,2),1)="/" THEN
      Formula$=MID$(BSpread$(CellCol,CellRow,2),2)
      Total=0:ErrSw=0
      IF Formula$="TOTCOL" THEN
        GOSUB TotalCols
      ELSEIF Formula$="TOTROW" THEN
        GOSUB TotalRows
      ELSEIF Formula$<>"*" THEN
        GOSUB Evaluate
      END IF
      IF LEN(STR$(Total))>CWidth THEN
        ErrMsg$="Result too big!":ErrSw=1
      END IF
      IF ErrSw=1 THEN
        BEEP:COLOR Mag,Blk:LOCATE 2,1:PRINT SPACE$(68)
        LOCATE 2,67-LEN(ErrMsg$):PRINT ErrMsg$
      END IF
    END IF
  NEXT
NEXT
GOTO CalcKit

TotalCols:
IF CellRow=1 THEN
  BSpread$(CellCol,CellRow,1)="0"
ELSE
  FOR y=1 TO CellRow-1
    IF LEFT$(BSpread$(CellCol,y,2),1)="/" THEN
      Total=Total+VAL(BSpread$(CellCol,y,1))
    ELSEIF LEFT$(BSpread$(CellCol,y,2),1)="/" THEN
      IF LEFT$(BSpread$(CellCol,y,1),1)="/" THEN
        Total=Total+VAL(BSpread$(CellCol,y,1))
      END IF
    END IF
  NEXT
END IF
BSpread$(CellCol,CellRow,1)=STR$(Total):GOSUB PrintCell
RETURN

TotalRows:
IF CellCol=1 THEN
  BSpread$(CellCol,CellRow,1)="0"
ELSE
  FOR x=1 TO CellCol-1
    IF LEFT$(BSpread$(x,CellRow,2),1)="/" THEN
      Total=Total+VAL(BSpread$(x,CellRow,1))
    ELSEIF LEFT$(BSpread$(x,CellRow,2),1)="/" THEN
      IF LEFT$(BSpread$(x,CellRow,1),1)="/" THEN
        Total=Total+VAL(BSpread$(x,CellRow,1))
      END IF
    END IF
  NEXT
END IF
BSpread$(CellCol,CellRow,1)=STR$(Total):GOSUB PrintCell
RETURN

Evaluate:
op$="*":CellValue$="" :Formula$=Formula$+"*"
FOR x=1 TO LEN(Formula$)
  x$=MID$(Formula$,x,1):y=INSTR(op$,x$)
  IF (x$="[" OR RelSw=1) THEN
    RelSw=1:CellValue$=CellValue$+x$
    IF x$="]" THEN RelSw=0
  ELSEIF (y=0) OR (y<>0 AND CellValue$="") THEN
    CellValue$=CellValue$+x$
  ELSE

```

```

y$=LEFT$(CellValue$,1)
IF (y$<CHR$(48) OR y$>CHR$(57)) AND (y$<>"-" AND y$<>".") THEN
  Cell$=CellValue$:GOSUB GetCellID
  IF AllOK THEN
    y$=BSpread$(TempCol,TempRow,1)
    z$=BSpread$(TempCol,TempRow,2)
    IF LEFT$(z$,1)="/" THEN
      CellValue$=y$
    ELSEIF LEFT$(z$,1)="/" THEN
      IF LEFT$(y$,1)="/" THEN
        CellValue$=y$
      ELSE
        ErrMsg$="Unresolved Formula at "+CellValue$:ErrSw=1
      END IF
    ELSE
      ErrMsg$="Character/Null at "+CellValue$:ErrSw=1
    END IF
  END IF
END IF
IF ErrSw<>1 THEN
  IF op$="+" THEN Total=Total+VAL(CellValue$)
  IF op$="-" THEN Total=Total-VAL(CellValue$)
  IF op$="*" THEN Total=Total*VAL(CellValue$)
  IF op$="/" THEN Total=Total/VAL(CellValue$)
  op$=x$:CellValue$=""
ELSE
  x=LEN(Formula$)
END IF
END IF
NEXT
IF ErrSw=0 THEN
  BSpread$(CellCol,CellRow,1)=STR$(Total):GOSUB PrintCell
END IF
RETURN

CalcKit:
LINE(609,9)-STEP(6,6),Mag,bf:Saved=0
GOTO WaitForUser

PRGoTo:
GOSUB GoToCell:GOTO WaitForUser

ClrSheet:
Prompt$=" OK to clear":Prompt2$=" Sheet?":GOSUB YesNo
IF NOT Yes THEN WaitForUser
COLOR Gra,Blk:LOCATE 2,1:PRINT"Clearing Sheet...":SPACE$(51)
FOR x=1 TO MaxCol
  FOR y=1 TO MaxRow
    BSpread$(x,y,1)=""
    BSpread$(x,y,2)=""
  NEXT
NEXT
LINE(40,32)-STEP(560,152),Gra,bf
LINE(609,9)-STEP(6,6),Blk,bf
Saved=-1:GOTO WaitForUser

ResetSheet:
Prompt$=" OK to reset":Prompt2$=" Sheet?":GOSUB YesNo
IF NOT Yes THEN WaitForUser
COLOR Gra,Blk:LOCATE 2,1:PRINT"Resetting Sheet...":SPACE$(50)
FOR CellRow=1 TO MaxRow
  FOR CellCol=1 TO MaxCol
    IF LEFT$(BSpread$(CellCol,CellRow,2),1)="/" THEN
      BSpread$(CellCol,CellRow,1)=STRING$(CWidth,"*")
      GOSUB PrintCell
    END IF
  NEXT
NEXT
LINE(609,9)-STEP(6,6),Blk,bf
Saved=0:GOTO WaitForUser

CSz5:
CSz10:
CSz15:
CSz20:
MENU 2,5,1:MENU 2,6,1:MENU 2,7,1:MENU 2,8,1
IF menuIT=5 THEN MENU 2,5,2:CWidth= 5:NumCols=14
IF menuIT=6 THEN MENU 2,6,2:CWidth=10:NumCols= 7
IF menuIT=7 THEN MENU 2,7,2:CWidth=15:NumCols= 4
IF menuIT=8 THEN MENU 2,8,2:CWidth=20:NumCols= 3
GOSUB DoColHdg:GOSUB ShoData:GOTO WaitForUser

TwoDec:
MENU 2,9,2:MENU 2,10,1:Mask$="#####.##"
GOSUB ShoData:GOTO WaitForUser

NoDec:
MENU 2,9,1:MENU 2,10,2:Mask$="#####-"
GOSUB ShoData:GOTO WaitForUser

HoldRlse:
IF NOT Holding THEN
  COLOR Gra,Blk:LOCATE 2,1:PRINT SPACE$(68)
  LOCATE 2,1:PRINT"How Many? ":
  Response$="":GetData Response$,"INT",1:HoldCols=VAL(Response$)
  IF HoldCols>NumCols THEN HoldCols=NumCols-1

```

(continued)


```

StartCol=StartCol+HoldCols:FirstCol=40+CWidth*8*HoldCols
NumCols=NumCols-HoldCols:PrintCol=6+CWidth*HoldCols
MENU 2,5,0:MENU 2,6,0:MENU 2,7,0:MENU 2,8,0
MENU 2,11,1," Release Cols ":Holdings=-1:GOSUB DoColHdg
MENU 1,0,0:MENU 2,12,0:MENU 4,6,0:MENU 4,8,0
ELSE
StartCol=StartCol-HoldCols
IF StartCol<1 THEN StartCol=1
FirstCol=40:PrintCol=6:Holdings=0
MENU 2,5,1:MENU 2,6,1:MENU 2,7,1:MENU 2,8,1
IF CWidth= 5 THEN MENU 2,5,1:NumCols=14
IF CWidth=10 THEN MENU 2,6,2:NumCols= 7
IF CWidth=15 THEN MENU 2,7,2:NumCols= 4
IF CWidth=20 THEN MENU 2,8,2:NumCols= 3
MENU 2,11,1," Hold 1st Cols "
GOSUB DoColHdg:GOSUB ShoData
MENU 1,0,1:MENU 2,12,1:MENU 4,6,1:MENU 4,8,1
END IF
GOTO WaitForUser

Template:
Prompt$=" OK to Make":Prompt2$=" a Template?":GOSUB YesNo
IF NOT Yes THEN WaitForUser
COLOR Gra,Blk:LOCATE 2,1:PRINT"Making Template...":SPACE$(50)
FOR CellRow=1 TO MaxRow
FOR CellCol=1 TO MaxCol
x$=LEFT$(BSpread$(CellCol,CellRow,2),1)
IF x$="N" THEN
BSpread$(CellCol,CellRow,1)=""
BSpread$(CellCol,CellRow,2)=""
GOSUB PrintCell
ELSEIF x$="F" THEN
BSpread$(CellCol,CellRow,1)=STRING$(CWidth,"*")
GOSUB PrintCell
END IF
NEXT
NEXT
LINE(609,9)-STEP(6,6),Blk,bf
Saved=0:GOTO WaitForUser

Kit:
IF NOT Saved THEN
Prompt$="Sheet not saved":Prompt2$="Quit 'BSpread'?":
GOSUB YesNo:IF NOT Yes THEN WaitForUser
END IF
MENU RESET:COLOR Gra,Blk
' $IGNORE ON
CLEAR ,25000
' $IGNORE OFF
CLS:END

YesNo:
WINDOW 2,,(120,51)-(258,97),0,-1
COLOR Blk,Yel:x=22
FOR n=32 TO 0 STEP -1
LINE(0,n)-STEP(137,0),Yel
x=x+1
LINE(0,x)-STEP(137,0),Yel
NEXT
LINE(15,31)-STEP(32,16),Blk,bf:LINE(95,31)-STEP(32,16),Blk,bf
LINE(12,28)-STEP(32,16),Gra,bf:LINE(92,28)-STEP(32,16),Gra,bf
LINE(12,28)-STEP(32,16),Mag,b:LINE(92,28)-STEP(32,16),Mag,b
LINE(13,29)-STEP(30,14),Mag,b:LINE(93,29)-STEP(30,14),Mag,b
LOCATE 2,2:PRINT Prompt$:LOCATE 3,2:PRINT Prompt2$
COLOR Mag,Gra:LOCATE 5,3:PRINT"Yes":PTAB(100);"No":
NoGdgt=-1
WHILE NoGdgt
WHILE MOUSE(0)=0:WEND
x=MOUSE(1):y=MOUSE(2)
IF y>28 AND y<44 THEN
IF x>12 AND x<52 THEN
Yes=-1:NoGdgt=0
ELSEIF x>92 AND x<122 THEN
Yes=0:NoGdgt=0
END IF
END IF
WHILE MOUSE(0)<>0:WEND
WEND
WINDOW CLOSE 2
RETURN

GoToCell:
Prompt$="Go To Cell? ":GOSUB GetCellNum
StartCol=TempCol>LastCol=600
IF StartCol+NumCols>MaxCol THEN
LastCol=(MaxCol+1-StartCol)*(CWidth*8))+40
END IF
StartRow=TempRow>LastRow=184
IF StartRow+19>MaxRow THEN LastRow=((MaxRow+1-StartRow)*8)+32
GOSUB DoColHdg:GOSUB DoRowHdg:GOSUB ShoData
RETURN

GetCellNum:
COLOR Gra,Blk:LOCATE 2,1:PRINT SPACE$(68):Cell$=""
GCNget:
LOCATE 2,1:PRINT Prompt$:
GetData Cell$,"CHAR",4:GOSUB GetCellID

```

```

IF NOT ALLOK THEN
COLOR Mag,Blk:LOCATE 2,67-LEN(ErrMsg$)
PRINT ErrMsg$:BEEP
COLOR Gra,Blk:GOTO GCNget
END IF
RETURN

GetCellID:
Cell$=UCASE$(Cell$):ALLOK=-1:ErrSw=0
IF LEFT$(Cell$,1)~="[" THEN
n=1:GOSUB RelCell
IF ALLOK AND MID$(Cell$,n,1)~="," THEN
GOSUB RelCell
ELSE
IF MID$(Cell$,n,1)~"]" THEN
ErrMsg$="Missing Col/Row: "+Cell$:ErrSw=1:ALLOK=0
ELSE
IF ErrSw=0 THEN GOSUB RelCell
END IF
END IF
ELSE
GOSUB AbsCell
END IF
RETURN

RelCell:
z$=MID$(Cell$,n,1)
IF z$~="[" OR z$~="," THEN
n=n+1:z$=MID$(Cell$,n,1)
IF z$="C" OR z$="R" THEN
n=n+1:a$=MID$(Cell$,n,1)
IF a$="+" OR a$="-" THEN
n=n+1:y$=MID$(Cell$,n+1,1)
IF y$="," OR y$="]" THEN
y$=MID$(Cell$,n,1):n=n+1
ELSE
y$=MID$(Cell$,n,2):n=n+2
END IF
IF z$="C" THEN
IF a$="+" THEN TempCol=CellCol+VAL(y$)
IF a$="-" THEN TempCol=CellCol-VAL(y$)
IF TempCol<1 OR TempCol>MaxCol THEN
ErrMsg$="Illegal Column: "+Cell$:ErrSw=1:ALLOK=0:TempCol=1
END IF
ELSEIF z$="R" THEN
IF a$="+" THEN TempRow=CellRow+VAL(y$)
IF a$="-" THEN TempRow=CellRow-VAL(y$)
IF TempRow<1 OR TempRow>MaxRow THEN
ErrMsg$="Illegal Row: "+Cell$:ErrSw=1:ALLOK=0:TempRow=1
END IF
END IF
ELSE
ErrMsg$="Illegal Operator: "+Cell$:ErrSw=1:ALLOK=0
END IF
ELSE
ErrMsg$="Must Specify a C or R: "+Cell$:ErrSw=1:ALLOK=0
END IF
ELSE
ErrMsg$="Illegal Relative Format: "+Cell$:ErrSw=1:ALLOK=0
END IF
RETURN

AbsCell:
TempCol=0
IF LEN(Cell$)<2 THEN ACerr1
IF LEN(Cell$)=2 OR LEN(Cell$)=4 THEN
n=LEN(Cell$)/2
TempCol$=LEFT$(Cell$,n):TempRow$=RIGHT$(Cell$,n)
ELSE
IF MID$(Cell$,2,1)<CHR$(58) THEN
TempCol$=LEFT$(Cell$,1):TempRow$=RIGHT$(Cell$,2)
ELSE
TempCol$=LEFT$(Cell$,2):TempRow$=RIGHT$(Cell$,1)
END IF
END IF
IF LEN(TempCol$)=2 THEN
IF LEFT$(TempCol$,1)<>RIGHT$(TempCol$,1) THEN ACerr2
TempCol=26:TempCol$=RIGHT$(TempCol$,1)
END IF
TempCol=TempCol+ASC(TempCol$)-64
IF TempCol<1 OR TempCol>MaxCol THEN ACerr2
TempRow=VAL(TempRow$)
IF TempRow<1 OR TempRow>MaxRow THEN ACerr3
GOTO ACexit
ACerr1:ErrMsg$="Illegal CellID: ":GOTO ACerrEnd
ACerr2:ErrMsg$="Invalid Column: ":GOTO ACerrEnd
ACerr3:ErrMsg$="Invalid Row: ":GOTO ACerrEnd
ACerrEnd:ErrMsg$=ErrMsg$+Cell$:ALLOK=0:ErrSw=1
ACexit:
RETURN

ShoData:
LINE(FirstCol,FirstRow)-(600,192),Gra,bf
rowtop=0:oldtop=0:colleft=0:oldleft=0
FOR CellRow=StartRow TO EndRow
FOR CellCol=StartCol TO EndCol
GOSUB PrintCell

```



```

NEXT
NEXT
CellCol=StartCol:CellRow=StartRow
IF CellMgmt THEN x=FirstCol+4:y=FirstRow+4:GOSUB MouseOK
RETURN

MouseOK:
n=CWidth*8
rowtop=INT(y/8)*8:colleft=INT((x-40)/n)*n+40
IF colleft>LastCol-n THEN colleft=LastCol-n
IF rowtop<oldtop OR colleft<oldleft THEN
  IF oldtop>0 THEN PUT(oldleft,oldtop),CellLite%
  PUT(colleft,rowtop),CellLite%
  oldtop=rowtop:oldleft=colleft
END IF
RETURN

PrintCell:
IF colleft>0 THEN PUT(colleft,rowtop),CellLite%
row%=PrintRow%+CellRow-StartRow
col%=PrintCol%+(CellCol-StartCol)*CWidth
IF row%<24 AND row%>4 AND col%<(77-CWidth) AND col%>PrintCol% THEN
  LOCATE row%,col%:COLOR Blk,Gra
  x$=BSpread$(CellCol,CellRow,1)
  y$=LEFT$(BSpread$(CellCol,CellRow,2),1)
  IF (y$="N" OR y$="F") AND (LEN(x$)>CWidth) THEN
    COLOR Mag:PRINT"LEN":STRING$(CWidth-5,"*"):COLOR Blk
  ELSE
    n=CWidth-LEN(x$):IF n<0 THEN n=0
    IF y$="C" OR y$="M" OR LEFT$(x$,1)="" THEN
      PRINT LEFT$(x$,CWidth):SPACE$(n):
    ELSE
      PRINT USING RIGHT$(Mask$,CWidth):VAL(x$):
    END IF
  END IF
END IF
END IF
IF colleft>0 THEN PUT(colleft,rowtop),CellLite%
RETURN

DoColHdg:
COLOR Blk,Mag:LOCATE 4,6:PRINT SPACES(70):LOCATE 4,PrintCol%
GET(24,24)=(24+CWidth*8-1,31),CellLite%
EndCol=StartCol+NumCols-1:IF EndCol>MaxCol THEN EndCol=MaxCol
FOR n=StartCol TO EndCol
  PRINT SPC(CWidth-3):ColHdg$(n):":":
NEXT
IF ColLite>0 THEN LINE(ColLite,17)-STEP(10,5),Blk,bf
n=41-13+(14*StartCol)
LINE(n,17)-STEP(10,5),Yel,bf:ColLite=n
RETURN

DoRowHdg:
COLOR Blk,Mag:row%=5
LINE(24,24)-STEP(15,160),Mag,bf
LINE(600,24)-STEP(15,160),Mag,bf
EndRow=StartRow+18:IF EndRow>MaxRow THEN EndRow=MaxRow
FOR n=StartRow TO EndRow
  LOCATE row%,4:PRINT USING "###":n:
  PRINT SPC(70):PRINT USING "###":n:
  row%=row%+1
NEXT
IF RowLite>0 THEN LINE(6,RowLite)-STEP(8,1),Blk,bf
n=30-3+(3*StartRow)
LINE(6,n)-STEP(8,1),Yel,bf:RowLite=n
RETURN

BasicError:
COLOR Mag,Blk:BEEP:ErrSw=1
IF Err=53 THEN
  LOCATE 2,53:PRINT"File Not Found!"
  RESUME AfterOpen
ELSE
  LOCATE 2,1:PRINT SPACES(68)
  LOCATE 2,1:PRINT"AmigaBasic Error:":ERR:
END IF
MENU 1,0,1:MENU 2,0,1:MENU 3,0,0:MENU 4,0,0
RESUME WaitForUser

Initialize:
FOR n=1 TO MaxCol:READ ColHdg$(n):NEXT
DATA "A","B","C","D","E","F","G","H"
DATA "I","J","K","L","M","N","O","P"
DATA "Q","R","S","T","U","V","W","X"
DATA "Y","Z","AA","BB","CC","DD","EE","FF"
DATA "GG","HH","II","JJ","KK","LL","MM","NN"
PALETTE 0,0,0:PALETTE 1,0,0,0
PALETTE 2,0,0,0:PALETTE 3,0,0,0
COLOR ,Blk:CLS
LINE(24,24)-STEP(592,160),Mag,bf
LINE(40,32)-STEP(560,152),Gra,bf
LINE(40,16)-STEP(560,7),Gra,b
LINE(4,29)-STEP(12,150),Gra,b
FOR n=54 TO 592 STEP 14:LINE(n,16)-STEP(0,7),Gra:NEXT
FOR n=32 TO 176 STEP 3:LINE(4,n)-STEP(12,0),Gra:NEXT
GET(24,24)=(31,31),IPcursor%
GOSUB DoColHdg:GOSUB DoRowHdg
COLOR Gra,Blk:LOCATE 2,69:PRINT"Changed:"

```

```

LINE(608,8)-STEP(8,8),Gra,b
COLOR Yel,Blk:LOCATE 1,1:PRINT"Current Sheet: unnamed"
LOCATE 1,69:PRINT"BSpread"
MENU 1,0,1,"Project Mgmt I"
MENU 1,1,1," Load (L)":MENU 1,2,1," Save (S)"
MENU 1,3,1," Save As (A)":MENU 1,4,1," Text File(T)"
MENU 1,5,1," Print (P)":MENU 1,6,1," Quit (Q)"
MENU 2, 0,1,"Project Mgmt II"
MENU 2, 1,1," Calculate (C)":MENU 2, 2,1," Go To Cell...(G)"
MENU 2, 3,1," Clear Sheet ":MENU 2, 4,1," Reset Sheet "
MENU 2, 5,1," Cell Size 5 ":MENU 2, 6,2," Cell Size 10 "
MENU 2, 7,1," Cell Size 15 ":MENU 2, 8,1," Cell Size 20 "
MENU 2, 9,2," Two Decimals ":MENU 2,10,1," No Decimals "
MENU 2,11,1," Hold 1st Cols ":MENU 2,12,1," Make a Shell "
MENU 3,0,0,"Cell Mgmt I"
MENU 3,1,1," Cell Data (D)":MENU 3,2,1," Cell Formula (F)"
MENU 3,3,1," Zero Cell ":MENU 3,4,1," Blank Cell "
MENU 3,5,1," Copy Cell ":MENU 3,6,1," Move Cell "
MENU 3,7,1," Propagate Right ":MENU 3,8,1," Propagate Down "
MENU 3,9,1," Go To Cell...(G)":MENU 3,10,1," Exit (X)"
MENU 4,0,0,"Cell Mgmt II"
MENU 4,1,1," Zero Column ":MENU 4,2,1," Zero Row "
MENU 4,3,1," Blank Column ":MENU 4,4,1," Blank Row "
MENU 4,5,1," Insert Column":MENU 4,6,1," Insert Row "
MENU 4,7,1," Remove Column":MENU 4,8,1," Remove Row "
COLOR Blk,Gra:LOCATE 19,32:PRINT"Click" to Start"
LOCATE 7,19:PRINT"BSpread" - an AmigaBasic(tm) Spread Sheet"
LOCATE 9,20:PRINT"Written for 'Amazing Computing' Magazine"
LOCATE 11,39:PRINT"by"
LOCATE 15,22:PRINT"Copyright (c) 1987 by FelineSystems"
LOCATE 16,31:PRINT"All Rights Reserved"
COLOR Mag,Gra:LOCATE 13,32:PRINT"Bryan D. Catley"
LOCATE 21,14:PRINT"2221 Glasgow Road Alexandria "
PRINT"Virginia 22307-1819"
PALETTE 0,0,0:PALETTE 1,.75,.75
PALETTE 2,1,0,1:PALETTE 3,1,.8,0
RETURN

SUB GetData (Text$,DataType$,maxlen%) STATIC
  SHARED TypeFlag$,IPcursor%
  WHILE INKEYS<>"":WEND
  Start=POS(0):Cur=0:xpix=(Start-1)*8:ypix=(CSRLIN-1)*8
  ShoText:
  IF LEN(Text$)=0 THEN TypeFlag$="N"
  n=maxlen%+1-LEN(Text$):IF n<0 THEN n=0
  PRINT Text$+SPACES(n):LOCATE ,Start
  xpix=(Start+Cur-1)*8:PUT(xpix,ypix),IPcursor%
  NxtChar:
  x$="":LeftPart$="":RightPart$=""
  WHILE x$=""&x$=INKEYS:WEND
  IF x$=CHR$(30) THEN CurRight ` Right-cursor
  IF x$=CHR$(31) THEN CurLeft ` Left-cursor
  IF x$=CHR$(8) THEN DelLeft ` Back-space key
  IF x$=CHR$(127) THEN DelRight ` Delete key
  IF x$=CHR$(27) THEN ClrText ` Escape key
  IF x$=CHR$(13) THEN GetDone ` Return key
  IF DataType$="CHAR" AND TypeFlag$<"C" THEN
    IF (x$<CHR$(48) OR x$>CHR$(57)) AND (x$<>".") THEN
      END IF
      IF LEN(Text$)>0 OR x$<>"-" THEN TypeFlag$="C"
    ELSEIF DataType$="INT" THEN
      IF x$<CHR$(48) OR x$>CHR$(57) THEN
        BEEP:GOTO NxtChar
      END IF
    END IF
  END IF
  InsertChar:
  IF LEN(Text$)=maxlen% THEN BEEP:GOTO NxtChar
  IF Cur>0 THEN LeftPart$=MID$(Text$,1,Cur)
  IF LEN(Text$)>0 THEN RightPart$=MID$(Text$,Cur+1,LEN(Text$)-LEN(LeftPart$))
  Text$=LeftPart$+x$+RightPart$:Cur=Cur+1:GOTO ShoText
  CurRight:
  IF Cur=LEN(Text$) THEN NxtChar
  Cur=Cur+1:GOTO ShoText
  CurLeft:
  IF Cur=0 THEN NxtChar
  Cur=Cur-1:GOTO ShoText
  DelLeft:
  IF LEN(Text$)=0 OR Cur=0 THEN BEEP:GOTO NxtChar
  IF Cur>1 THEN LeftPart$=MID$(Text$,1,Cur-1)
  IF LEN(Text$)>Cur THEN RightPart$=MID$(Text$,Cur+1,LEN(Text$)-Cur)
  Text$=LeftPart$+RightPart$:Cur=Cur-1:GOTO ShoText
  DelRight:
  IF LEN(Text$)=0 OR Cur=LEN(Text$) THEN BEEP:GOTO NxtChar
  IF Cur>0 THEN LeftPart$=MID$(Text$,1,Cur)
  IF Cur+1<LEN(Text$) THEN RightPart$=MID$(Text$,Cur+2,LEN(Text$)-Cur+1)
  Text$=LeftPart$+RightPart$:GOTO ShoText
  ClrText:
  PRINT SPACES(maxlen%+1):LOCATE ,Start
  Cur=0:Text$="":GOTO ShoText
  GetDone:
  PUT(xpix,ypix),IPcursor%
END SUB

```


Roomers

by The Bandito

An 80386 based BridgeBoard for the Amiga 2000???

The new "newfs" file system for Amiga hard disks is still in beta testing. Testing has shown that some Amiga programs circumvent the AmigaDOS file structure and functions to gain speed. These programs are incompatible with the faster file system. There are several commercial programs that do not work with the new software. The new software will reportedly need a new Kickstart and Workbench. It will be able to boot from any arbitrary device on the system, such as a hard disk or network connection.

Commodore is semi-officially working on version 1.3 of AmigaDOS. Technicians and programmers have been visiting the computer networks, asking "techies" for their suggestions for improving the operating system. The new software would need to be burned into ROMs for all existing Amiga 500 and 2000 owners, which means a trip to the local service center for a chip upgrade to run new software that might require the new DOS. Chances are that most software will continue to adhere to the 1.2 standard, given the great number of machines out there with that version of the operating system.

Rumors say the new "fatter Agnes" graphics chip that addresses one megabyte of CHIP memory has the ability to address two megabytes, too. No current Amiga design can handle it, though. The decision to address one meg or two is a "bond-out" option on the chip, meaning the choice

between one and two meg is made at the time when the silicon chip is placed in the ceramic chip package. The two meg design can't run in today's machines without a significant board design change.

This "bond-out" option also means Commodore can gear up for a new architecture on a moment's notice. Word has it that the Amiga 3000 design will have yet another bus design, but some slots would be backwards compatible to the current Zorro II design.

Rumor has it that Commodore is holding back on shipment of Amiga 2000s because they want to ship a two megabyte memory card and a hard disk controller with every machine.

Both the A2000 version of the Genlock peripheral and the A2000 composite adapter provide composite video. Commodore may never produce a separate composite adapter for the 500 and 2000, but you can always buy a third party composite adapter, and get a genlock to boot.

It turns out the coprocessor slot in the Amiga 2000 is actually compatible with original 86-pin spec expansion cards. The gender is wrong, but with an adapter, you can run one—and only one—old-style card in the 2000. You can't run any new Zorro style cards at the same time, however.

The Software Distillery is porting NetHack 2.x to the Amiga. NetHack is a multi-user game of Hack. They are also considering their own hard

disk software to increase the speed of the AmigaDOS. They still have driver code left from their ill-fated hard disk commercial venture.

At COMDEX, IBM showed OS/2 running on a Model 80, but it was crashing a lot during the demonstrations. If you looked closely, the revision dates on the software revealed that it was last modified the day before!

Recent talk says Commodore West Germany is working on a 80386 based BridgeBoard for the Amiga 2000. Rumors say it will be shown at a spring computer show.

More indications that Ashton-Tate is developing programs for the Amiga come from spies on bulletin boards close to A-T's main offices. Employees have been visiting local Amiga BBSs, looking for technical tips. Chances are they are porting the dBase data base program to the Amiga.

Rumors are flying that an Amiga hardware company is making a do-it-yourself Super Amiga with a 68020, a 68881, and lots of memory and card slots. It is made from the company's motherboard and an Amiga 500. You disassemble the Amiga 500, remove the custom chips, and drop them into the new circuit board. Other hardware companies are talking about expansion boxes for the Amiga 500, to turn it into a high-powered, 68020 based workstation for universities and software companies.

•AC•

As I See It

by Eddie Churchill

Another Software Generation is Born

First, Some Current Events

Well, the big move is over. Good-bye Charleston, hello Dallas. And with the new job and new home came some new equipment. Ole Faithful has been given an infusion of new life through a 2 meg Alegra board and a Phoenix hard drive. I also have a A2000 with 3 meg of memory and a 20 meg hard drive. I will report on how they all work, as I use and abuse them. Meanwhile, now that I've brought you up to date, let's move on to more important issues.

East is East, West is West, and Now the Twain Shall Meet!

Not very long ago, the only programs which worked together were written by the same company. New companies soon sprang up, making their livings by selling add-on programs which worked with other, "big ticket" programs. The demand for these types of programs grew into an entire market.

As both computers and users have grown more sophisticated, the desire to share "resources" has also grown. Nobody wants to re-invent the wheel. Once you are accustomed to a nifty feature, you sure hate to give it up. At the very end of this historical jaunt comes the Amiga with its multi-tasking operating system, making it absolutely necessary for programs to live together in peace. Peaceful co-existence does not mean file compatibility, though. IFF was *supposed* to provide that peace.

The problem was that when the IFF standard was decided upon, no one thought about animation. So, when desktop video emerged as the Amiga's strong suit, it caught IFF people off-guard. When a standard finally arose from all the confusion, several very good programs were left out in the cold.

Products are now appearing to solve this situation. Like the add-ons of old, products like InterChange can rectify some incompatibilities of older, still useful programs. Of course, programs that read other formats besides their own should always be encouraged. City Desk's inclusion of WordPerfect format is an example. Let's give a hearty "Well done!" to the good folks at Sunrize.

Don't rest on your laurels yet, gang. I'm still disappointed when features are only partially implemented. City Desk only understands the attribute and justification information from within a WordPerfect file. No tabs! No columns! Sigh. Sunrize isn't the only programming house to go only part of the way, however. How many desktop publishing packages or word processors which allow the inclusion of graphics also allow HAM? Aren't HAM pictures graphics? Sure, it would mean more work, but still...

Anyway, I know an upgrade of City Desk is going to allow importation of more info from the WordPerfect file, and the battle for the desktop publishing buck is just beginning to heat up. Look for some spiffy features in new packages in the near future. Isn't competition great?

Another Software Generation is Born!

First, there were Deluxe Paint and Images. (I suppose I have to include Graphicraft in here, but since it was still-born, I won't mention it again.) These first generation painting programs were, by today's standards, pretty crude. No dithering, no perspective, and no worky-worky with extra memory. It wasn't very long before the second generation appeared to make our lives easier. Once again, healthy competition gave us more features and better user-interfaces. Deluxe Paint II, Digi-Paint (filling in the gaps in DPII), and Prism all pushed the Amiga a little harder, while allowing us to do more than ever.

The third generation of paint programs has now arrived. Photon Paint, HAM's answer to Deluxe Paint II, arrived at work for me to beta-test. By the time you read this bit of wisdom, Photon Paint probably will be all the rage, and everything I say will be old hat. I'm going to put in my two cents' worth anyway. I think it's awesome—as evolutionary a paint program as Faery Tale Adventure was a game. MicroIllusions does it again! Its HAM features include texture mapping on any surface, some real spiffy brush manipulation (like twisting), and a simple to use perspective which makes DPII archaic! Photon Paint is truly the next generation. The rest of the graphic houses have to get *really* busy to catch up to Photon Paint.

(continued)



"Friendly advice - Knowledgeable staff"

"Hundreds of AMIGA products in stock!"

AMIGA™

"We specialize in AMIGA and C64/128!"

Now In Stock!

Insider 1-meg board • w/ Clock-Calendar

Call For Our Low Pricing!

New & Expanded

SOFTWARE

3670 Delaware Ave.
Kenmore, N.Y. 14217



SUPERMARKET

(716)873-5321

Rumor has it that EA is already taking steps to do just that, by acquiring D'Buddy. D'Buddy has evolved quite a bit from the days when it was just going to be a Gizmoz for graphic work. Now John Botteri at Digital Creations tells me it's a full HAM painting program in its own right. His description sounded really great, but no working copy has arrived for evaluation. Maybe EA can see into the future. (Remember when they claimed to "have vision?") Could D'Buddy become DPMII? Only time will tell.

Viruses and other Pestilence

The favorite topic on many bulletin boards nowadays is viruses. Many programs supposedly have viruses (especially if they are from Europe). We used to say software was "buggy." Now we say it has a virus. I guess next year programs will have bacteria...

I understand there really are some problems along these lines, but a lot of the talk sounds bogus to me. I think the computer industry as a whole has a tendency to give too much credence to fables. Look how we react when software turns out to be "vaporware." From what I can tell, the only real pests in the software industry are those who have software they didn't buy. They deserve whatever they get.

And Speaking of Europe...

How about those European arcade games? Some of them look familiar, don't they? Talk about look and feel! Garrison ought to be called Gauntlet II. If Amegas isn't an Arkanoid clone, I don't know what is. I'm all for competition and free enterprise, but how about some originality in arcade games? Why is everybody copying everybody else, so we get three versions of the same game? There are many arcade games that haven't been made for the Amiga yet, gang. If you need a memory jog, here are a few titles I'd like to see brought over:

Venture—Let's try some simple arcade games. Most games today are just too hard!

M.U.L.E.—I will never forgive EA for not bringing this out on the Amiga. Come on, Europeans!

Impossible Mission—If the C-64 can have it, why can't the Amiga?

Boulder Dash—Where is Emerald Mines, anyway?

I, Robot—Hey, it was done on an Amiga first anyway. How tough can it be? So what if Atari has the rights. Clones are in style, right?

Arcade Classics—How about three to a set? Volume 1 could be PacMan, Donkey Kong, and Dig Dug. Volume 2 could be Defender, Gorf, and Berzerk.

Wizardry—If Infocom can put all three Zorks on a disk, Sir-Tech can get at least two. If they won't, which is likely, vector-drawn dungeons aren't copyrighted, you know.

See how easy? All it takes is a little imagination and a good memory.

The Mandatory Disclaimer

Well, time for me to scroll out of here. The preceding comments are products of my own fertile imagination. My new employer has nothing to do with this column. Neither do my friends (both of them), acquaintances, or colleagues. If your own thoughts are along these same lines, then you are as vector-oriented as I am.

•AC•

68000 Assembler Language Programming

on the Amiga™

Create a graphic image with the basic, low-level graphics routines in the ROM Kernel

by Chris Martin

Last time, we discussed the various structures used with the Amiga graphics routines in the ROM Kernel. This month I would like to present an assembly program that creates a multi-colored screen without using the Intuition routines. Remember, you can create a graphic image with the basic, low-level graphics routines in the ROM Kernel. Those routines are divided into two categories: display routines and drawing routines. An explanation of the order in which display routines must be called is essential.

Display Routines

Notice the space allocated for the structures at the end of the program. Certain structure elements use bytes, some use words, and others use long words of storage. For each element, I defined the necessary space to correctly align the structure's elements in memory. Each structure's name is listed as a label. The memory address of this label is necessary when you use any structure with the system routines. Each following element must be properly sized, since the system routines look for the value of each element in an offset from the address of the structure's start address. Certain elements of each structure have labels attached to make data storage into each element an easy task. Space is allocated using the syntax:

```
<label> ds.<size> .l .w .b> <# of longs, words, or bytes>
```

After space has been allocated for each structure, display routines must be called to create and display the structures' definitions. Examine the section of the program labeled "createscreen:." This section actually calls the display routines to create the display. (See Figure 1.) First, "createscreen" allocates memory for use by the bitplanes. Each is 320 bits wide by 200 bits tall (full-screen).

Next, the system routine AllocRaster is called, since the memory allocated must be in the first 512K of memory (CHIP memory). The Amiga's custom chips can only use memory in the first 512K. AllocRaster is called three times, once for each bitplane used by the bitmap. The pointer to the start-address of each bitplane is returned by AllocRaster in the D0 register.

Now, the pointer is stored in the section of the bitmap structure labeled "plane<n>" (<n> is a number from 1 to 8, representative of the bitplane number). Next, the structures are initialized. The default values of each element are stored in their memory spaces. At any time, we can change the values of each element, simply by storing a value in a data register, and then storing the data register in the address of that element's label. For example, to store a 3 in the bitmap element defining depth, do the following:

```
move.b #3,d0
move.b d0,bmdepth
```

"bmdepth" is the label of that element. The following are the routines called for initializing the structures, in the order they are called:

InitView()	for initializing the View structure.
InitVPort()	for initializing the ViewPort structure.
InitBitMap()	for initializing the BitMap structure.
InitRastPort()	for initializing the RastPort structure.

When structures are given space in memory, that space is sometimes filled with garbage. When a structure is initialized, it is filled with default data valid to a system routine which uses the structure. The above routines simply initialize structures, but without them, the program would crash.

The Copper

Before I discuss the routines which actually show the display, an explanation of the Copper is necessary. Copper stands for Coprocessor. It is a helper chip to one of the Amiga's custom chips called the Blitter. The Copper handles display, while the Blitter controls drawing and areafilling. The Copper is a computer in itself; it has its own instruction set, and it is constantly executing small programs in the Amiga's memory. To control the display, the Copper looks to a "list" of its instructions and executes them one at a time. The Copper instruction set manipulates the display based on "raster interrupts."

(continued)

AC/FORTRAN™

Mainframe quality, full feature ANSI FORTRAN 77 compiler includes: **Debugger**, **Linker**, **Library Manager**, **Runtime Library**, **IEEE math**, and **C interface**. Supports **Complex numbers**, **Virtual arrays**, **Overlays** and **Linking**. Not copy protected. \$295.

Version for CSA 68020/68881 Turbo board also available \$495.

AC/BASIC™

From the authors of **Microsoft BASIC** compiler for Macintosh, comes AC/BASIC for the Amiga. **Compatible with the Amiga BASIC** interpreter: has more features and includes **BLOCK IF**, **CASE** statement, and **STATIC** keyword extensions and executes up to **50x** faster. AC/BASIC is the new BASIC reference for MC68000 based personal computers. Not copy protected. \$195.

absoft

Scientific/Engineering Software

2781 Bond Street, Auburn Hills, MI 48057/(313) 853-0050

Amiga trademark of Commodore/Amiga. Microsoft trademark of Microsoft Corp.



Telephone orders welcome

Creating the Display

Finally, the routines that actually create the display are called in the following order:

MakeVPort()	Sets-up the ViewPort structure for display.
MrgCop()	Merges the new display specs with the existing display.
LoadView()	Shows the display.

MakeVPort creates a Copper program based on your screen characteristics. If you have three ViewPorts (See Figure 2.), each with a different color set or resolution, a Copper program is created to interrupt the Amiga when it must change the color set or modify the resolution. MrgCop works in conjunction with MakeVPort. It takes the Copper program created by MakeVPort and merges it with the existing Copper program. Finally, LoadView ties everything together and instructs the Copper to begin execution of the new Copper list.

Drawing Routines

All drawing routines use a very important structure called RastPort. The RastPort structure is directly tied to the bitmap, since all writing to the display is done into the bitmap. Here is the RastPort structure:

Structure:	RastPort	
Include File:	Graphics/RastPort.i	
Size	Components	Explanation
LONG	rp_Layer	Pointer to first layer
LONG	rp_BitMap	*** Pointer to BitMap structure ***
LONG	rp_AreaPtrn	Pointer to areafilling pattern
LONG	rp_TmpRas	Pointer to temporary space
LONG	rp_AreaInfo	Information for areafilling
LONG	rp_GelsInfo	Graphics ELeментS (animation)
BYTE	rp_Mask	Graphics bit masking
BYTE	rp_FgPen	Foreground pen #
BYTE	rp_BgPen	Background pen #
BYTE	rp_AOLPen	Pen # for outlining areafills
BYTE	rp_DrawMode	Drawing mode
BYTE	rp_AreaPtSz	Areafilling pattern size
BYTE	rp_Dummy	??? who knows ???
BYTE	rp_Linepatcnt	# of bytes used for line pattern
WORD	rp_Flags	Flags for the RastPort
WORD	rp_LinePtrn	Pointer to line pattern
WORD	rp_cp_x	Current pen position x
WORD	rp_cp_y	Current pen position y
8 LONG	rp_minterms	Special bit-plane manipulation
WORD	rp_PenWidth	Pen width
WORD	rp_PenHeight	Pen height
LONG	rp_Font	Font used currently in the RastPort
BYTE	rp_AlgoStyle	Algorithmic style (bold, italic)
BYTE	rp_TxFlags	Text flags
WORD	rp_TxHeight	Text height
WORD	rp_TxWidth	Text width
WORD	rp_TxBaseLine	Text baseline definition
WORD	rp_TxSpacing	Special Text spacing
WORD	rp_RP_User	Current RastPort user
14 LONG	rp_wordreserved	reserved space (system use)
8 LONG	rp_longreserved	
8 LONG	rp_reserved	

Remember, a raster display is created by an electron gun inside the monitor which the display's highlights dots (pixels) one row at a time. The gun moves down each row fast enough to draw the entire display in 1/60th of a second. In a raster interrupt, the computer says, "Wake me up when we get to row XXX." When the monitor has drawn row XXX, the computer is "interrupted," and a routine to manipulate the screen is called. When this routine finishes, the computer returns to its normal functions. The Copper instruction set is nothing compared to the 68000, but it is important and powerful, nonetheless. Here are the instructions:

- **WAIT** Wait for the raster beam to reach a specific screen position.
- **MOVE** Move data into one of the custom chip registers.
- **SKIP** Skip to the next instruction if the raster beam has already reached a specific screen position.

When defining a screen and calling the routines to display it, the Amiga creates a Copper program for you—how nice!

All we really need to define is the pointer to the bitmap; the rest is defined by system routines.

Drawing routines called by this program are as follows:

Move()	Move the pen to a coordinate in the bit-map
Draw()	Draw to a coordinate in the bit-map
RectFill()	Draw a filled rectangle
SetAPen()	Set the current pen color
SetRast()	Fill the entire bit-map with a color
SetDrMd()	Set the drawing mode

The drawing routines, as you will notice, are called in the same manner as the display routines. First, you must place the address of the Graphics Library ("graphicsbase") in the address register A6. The parameters are then passed through other address or data registers. For example, to set the bit-map to color #1, use the following syntax:

```
move.l graphicsbase,a6
lea.l rastport,a1      * "rastport" is the label of the structure
move.b #1,d1           * pen number 1
SYS SetRast(a6)        * "SYS" is macro to call a sys routine
```

A list of all routines (drawing and display) is in the ROM *Kernel Manual: Libraries and Devices*, available through Addison-Wesley. This manual provides a thorough explanation of all routines and their usage in C and assembly.

Next month, I'll discuss variations in on-screen display. I'll also introduce smooth scrolling and begin a discussion on dual-playfields. [Part II of ASSEMGR.ASM will appear next month - Ed]

Listing One

```
*****
*
* ASSEMGR.ASM Part I
*
* Assembly program to create a multi-color, non-intuition
* display.
*
* By Chris Martin, for Amazing Computing
*
* Compile with the following commands:
*
* 1> assem assemgr.asm -o assemgr.o -c w40000
* 1> alink assemgr.o to assemgr library amiga.lib
*
*****
* INCLUDE files
*
* INCLUDE "exec/types.i"
* INCLUDE "exec/funcdef.i"
* INCLUDE "graphics/gfx.i"
* INCLUDE "graphics/view.i"
* INCLUDE "graphics/rastport.i"
*
* Macros
*
EXT_SYS MACRO
XREF _LVO\1
ENDM

SYS MACRO
JSR _LVO\1
ENDM
```

(continued)

Looking For More "Byte" For Your Dollar?



Here is the answer, the model **PPI-1000** Expansion Unit for your Amiga 1000 by **Palomar Peripherals**.

- ☐ Full auto configuration
- ☐ Meets Zorro specifications
- ☐ Amiga buss pass-thru
- ☐ Built in power supply
- ☐ Power on/off controlled by Amiga
- ☐ Real time clock with battery backup
- ☐ Fast RAM
 - 2,4,6 or 8 megabytes
 - No wait states
- ☐ Disk drives
 - Supports up to 2 drives
 - 20 or 40 megabyte hard drive(s)
 - 3½" or 5¼" floppy drive
 - Fast DMA interface

With 20 meg hard drive and 2 meg RAM price is \$1395 plus shipping and handling (Calif. Res. add 6% sales tax). Requires Amiga DOS 1.2. For further information call **(619) 748-1202** or write to **Palomar Peripherals**.

PALOMAR PERIPHERALS

14580 High Pine
Poway, Ca 92064

* Here, I list the routines that this program calls.

```
EXT_SYS  OpenLibrary      * (libraryname,version) (A1,D0)
EXT_SYS  Move             * (rastport,x,y) (A1,D0,D1)
EXT_SYS  Draw             * (rastport,x,y) (A1,D0,D1)
EXT_SYS  SetAPen          * (rastport,color#) (A1,D0)
EXT_SYS  SetDrMd          * (rastport,drawmode) (A1,D0)
EXT_SYS  SetRGB4          * (viewport,color#,r,g,b)
                        * (A0,D0,D1,D2,D3)
EXT_SYS  SetRast          * (rastport,color#) (A1,D0)
EXT_SYS  InitView         * (view) (A1)
EXT_SYS  InitVPort        * (viewport) (A0)
EXT_SYS  InitBitMap       * (bitmap,depth,width,height)
                        * (A0,D0,D1,D2)
EXT_SYS  InitRastPort     * (rastport) (A1)
EXT_SYS  AllocRaster      * (width,height) (D0,D1)
EXT_SYS  MakeVPort        * (view,viewport) (A0,A1)
EXT_SYS  MrgCop           * (view) (A1)
EXT_SYS  LoadView        * (view) (A1)
EXT_SYS  RectFill         * (rastport,x1,y1,x2,y2)
                        * (A1,D0,D1,D2,D3)
```

External References

XREF _AbsExecBase * The Base of the Exec Library

Start of the program

start:

```
move.l  sp,savesp      * save the current stack pointer
jsr     openlibraries  * jump and return from 'openlibraries'
jsr     createscreen   * create the display
jsr     setcolors      * set the colors for the display
jsr     drawsomething  * draw some boxes, lines
```

* this is an infinite loop

infinite:

```
jmp     infinite      * isn't that original?
```

Open Libraries

openlibraries:

* Graphics Library

```
move.l  _AbsExecBase,a6
lea.l   graphics,a1
moveq   #0,d0
SYS     OpenLibrary(a6)
move.l  d0,graphicsbase * get base address of graphics lib
rts     * return
```

Create the screen

createscreen:

* allocate memory for the 3 bitmap planes

```
move.l  graphicsbase,a6
move.w  #320,d0
move.w  #200,d1
SYS     AllocRaster(a6)
move.l  d0,plane1
```

```
move.l  graphicsbase,a6
move.w  #320,d0
move.w  #200,d1
SYS     AllocRaster(a6)
move.l  d0,plane2
```

```
move.l  graphicsbase,a6
move.w  #320,d0
move.w  #200,d1
SYS     AllocRaster(a6)
move.l  d0,plane3
```

```
move.l  graphicsbase,a6
lea.l   view,a1
SYS     InitView(a6) * get address of the View
                        * Initialize the view
```

```
lea.l   viewport,a0
move.l  a0,view * pointer to first viewport in linked list
```

```
move.l  graphicsbase,a6
lea.l   viewport,a1 * get address of the ViewPort
SYS     InitVPort(a6) * Initialize the viewport
```

```
lea.l   rasinfo,a0
move.l  a0,vprasinfo * pointer to rasinfo structure
move.w  #320,d0 * viewport width
move.w  d0,vpwidth
moveq   #0,d0 * clear d0
move.w  #200,d0 * viewport height
move.w  d0,vpheight
moveq   #0,d0 * viewport modes ( 0 for 320x200 graphics )
move.w  d0,vpmodes
```

```
move.l  graphicsbase,a6
lea.l   bitmap,a0
move.b  #3,d0
move.w  #320,d1
move.w  #200,d2
SYS     InitBitMap(a6) * Initialize the BitMap
```

```
lea.l   bitmap,a0
move.l  a0,ribitmap
```

```
move.l  graphicsbase,a6
lea.l   view,a0 * View structure in a0
lea.l   viewport,a1 * viewport structure in a1
SYS     MakeVPort(a6) * call the MakeVPort routine
```

```
move.l  graphicsbase,a6
lea.l   view,a1 * View structure in a1
SYS     MrgCop(a6) * merge new coproc list with the old
```

```
move.l  graphicsbase,a6
lea.l   view,a1 * View structure in a1
SYS     LoadView(a6) * now show the display
```

```
lea.l   bitmap,a0
move.l  a0,ribitmap
```

```
move.l  graphicsbase,a6
lea.l   rastport,a1
SYS     InitRastPort(a6) * initialize the rastport structure
```

```
lea.l   bitmap,a0
move.l  a0,rpbitmap
```

rts

Set the colors

***** NOTE *****

* We are only going to set 4 color regs (for space purposes)
* you can cont setting the other 4 (8 in all) if you wish to
* use them.

setcolors:

```
move.l  graphicsbase,a6
lea.l   viewport,a0
move.l  #0,d0 * color # 0, set to R=0, G=0, B=0 (black)
move.l  #0,d1
move.l  #0,d2
move.l  #0,d3
SYS     SetRGB4(a6)
```

```
move.l  graphicsbase,a6
lea.l   viewport,a0
move.l  #1,d0 * color # 1, set to R=4, G=10, B=15
move.l  #4,d1
move.l  #10,d2
move.l  #15,d3
SYS     SetRGB4(a6)
```

```
move.l  graphicsbase,a6
lea.l   viewport,a0
move.l  #2,d0 * color # 2, set to R=15, G=0, B=5
move.l  #15,d1
move.l  #0,d2
move.l  #5,d3
SYS     SetRGB4(a6)
```

```
move.l  graphicsbase,a6
lea.l   viewport,a0
```


Hilton Android's ROBOT READERS

By Don Hicks

In today's fast and hurried world, it is possible to forget the needs of young readers. It is often easy to replace the reading experience of parent and child with a complacent attachment to television. For years, we have been told the computer would herald a new approach to an old problem, the one-on-one needs of a new reader. Hilton Android has given the Amiga community several short, interesting computerized stories just to help new readers.

Robot Readers currently consists of four titles with a fifth due soon: Chicken Little, Three Little Pigs, Aesop's Fables, Little Red Hen, and soon, The Ugly Duckling. Each title comes on a single, brightly colored disk with an extremely short two page instruction book. The entire emphasis of Robot Readers is keeping the program simple. A small child could be left alone with the Amiga (all varieties) and be able to play and learn on his own.

The disk is input at the Workbench prompt and then takes over the machine. (Obviously multi-tasking was not a consideration for young readers.) The young reader is presented with a title screen and menu bar. This menu bar appears on every screen throughout the session, and is the only means of controlling the program. By utilizing this simple menu structure, robot readers is interfaced entirely with the mouse. The young reader does not need to type or search the keyboard in any way.

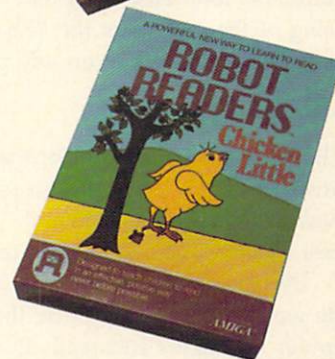
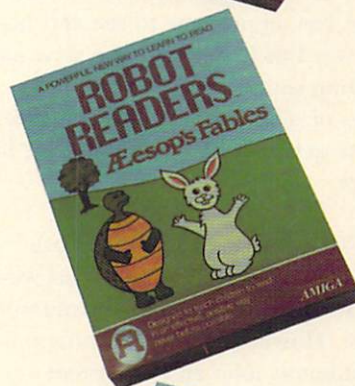
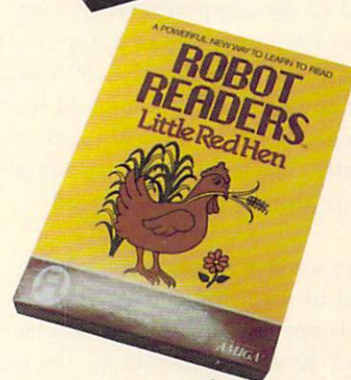
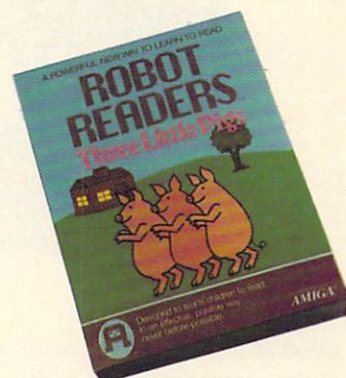
The menu consists of:
G READ F S PAGE

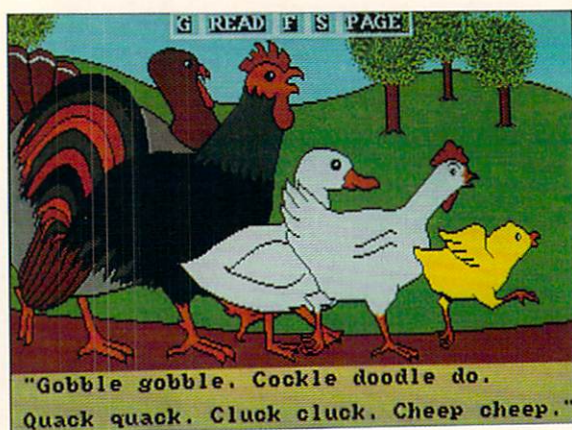
"G" stands for game. Positioning the arrow cursor over the G, then pressing either mouse button, places the program in a game mode for the current page. The game consists of finding the word requested by the computer, "Find the word...." The young reader positions the pointer over the correct word and presses either mouse button.

If the correct word is found, the word is highlighted by a second primary color, recited by the computer, changed to a third color, and a new word is requested by the computer. As each new word is requested and found without mistakes, it turns to the third color. This process continues until the entire page of text is the new color. The page flashes once, then stops. The reader is now able to continue.

If the reader does not hear the word correctly before pressing an incorrect word, he can repress the "G," and the word will be repeated. If he presses an incorrect word at any time in the game, the word is highlighted, spoken, and the game stops. The player must repress the "G" to begin again.

When pressed, "READ," reads the page in standard Amiga phonetics. There has been no attempt to create digitized sounds or voices for the series. Certain voices are created for characters by altering the Amiga voice in pitch, in the same manner used in "speech toy."





"F" is selected to increase the reading speed of the computer. With each press of the mouse button, the reading speed is increased by 25 to 50 words per minute. The highest speed level of speech is a screeching 400 words per minute. The child would no longer require these aids at that speed.

"S" is selected to slow the reading speed of the computer by 10 to 50 words per minute. At the slowest speed, 40 words per minute, the new reader has ample time to see and hear each word as it is read. The slow and plodding sound soon gets on the nerves of older readers and is a sure way to get a set of headphones for the Amiga.

"PAGE" advances you to the next page. Unfortunately, there is no way to move back either single or multiple pages. However, the entire program is a continuous loop and consistent pressing of the mouse on the "READ" prompt places the reader on any page he wants.

Not listed in a menu, but addressed in the instructions, is the ability to select individual words on a page and request the computer to "sound them out." By placing the cursor over a single word and pressing once, the computer highlights and reads the single word. By selecting the same word again, the computer highlights the word, reads it by syllables (sometimes its own variety of syllables), then reads it by the individual letters or

groups of letters that produce the individual sounds. Finally, the computer reads the entire word very slowly. This is designed to build a better understanding of the group of letters that produce a particular sound in a particular word.

Possibly the most immediately appealing portion of the program is the graphics. Each page quickly appears on the screen with its text. The pages are not unlike the small readers most of today's computerists read as children. Bright colors and uncomplicated drawings set the stage for each page of text.

Drawbacks

Since the standard Amiga speech interface was used for convenience, some faults occur in the reading portion and the syllable portion of the program. Certain words, such as "animals" and "tell," sound funny and stilted. Other words, such as "summer," develop new, interesting syllable structures. Also, digitized sounds and voices would have added a great deal of life to the material.

Some of the vocabulary words are above a beginning reader's comprehension level. However, great care has been taken to allow the reader to select the individual words and understand their meanings through the context of the sentence.

Strong Points

The program is easy to use and is clearly written for a young person to use unattended. The brightly colored illustrations, simple mouse interface, endless loop program construction, and simple games allow a young reader to progress at his own pace and still remain interested in the story.

A Personal Bias

This program is quick and simple and gives the new reader some support. However, it should only be used to augment and reinforce a young reader's learning. The gift of reading is a personal value, passed from parent to child. Through reading, we share the joys and understandings which cross generations and time. There will never be a substitute for the time spent with a child learning a story that is old for you and new for him. While we teach them the basics of a new talent, they show us a new and remembered view of the world.

Hilton Android has done an admirable job of producing a vehicle for self-learning and reading practice. Yet, an adult is still required to add the depth and questioning necessary to broaden the understanding of the reader. As an example, *Chicken Little* tells a great story, however, the moral is not available in the text, only through the questioning of the reader. Hilton Android can help with the mechanics, but a parent is still necessary for the magic.

Robot Readers: \$29.95 each
Chicken Little
Three Little Pigs
Aesop's Fables
Little Red Hen
 and soon.....
The Ugly Duckling

Hilton Android
 P.O. Box 7437
 Huntington Beach, CA 92615-7437
 (714) 963-4584

Modula-2 Programming

on the Amiga™

M2Amiga by Interface Technologies

[Steve took a step back from CALC this month to give you a chance to input all the listings. Some of the listings were carried over to this issue, and the remaining ones will be listed next month. Steve will continue with CALC next month. - Ed]

There is a new contender in the Amiga Modula-2 arena. The newcomer is M2Amiga, distributed in the United States by Interface Technologies Corporation of Houston. This compiler was developed in Switzerland by the people who worked with Niklaus Wirth at ETH Zurich; thus, it reflects philosophies newer than those present when the TDI and Benchmark compilers were developed.

What You Get

The package consists of one disk, a manual, and the book *Modula-2: A Seafaring Guide and Shipyard Manual* by Joyce. The disk contains a README file, an editor (based on Emacs), the compiler, the linker, a couple of utility programs, some definition files, and some demo programs.

One of the utilities (called DOME) sets up a working bootable disk that contains all the necessary executables. All you need is a blank formatted disk and a workbench disk. The workbench disk is necessary for copying all workbench related files, since the disk supplied does not have the workbench on it. This setup program is quite slow and causes the disk drive to grind a lot (It took over 10 minutes to run.), but very little manual intervention is required, so there's not much chance for an inexperienced user to mess things up.

The Environment

The second utility program, called M2Project, is an interesting one; it lets the programmer/user set up "projects." A project is a directory which contains at least two other subdirectories: one called SYM and the other called OBJ. The intention is that the user keeps all work related to a given program in such a project, and all the related symbol files (the result of compiling definition modules) are saved in the SYM directory, while all object files (the result of compiling implementation modules) are stored in the OBJ directory.

Sometimes it is necessary to refer to symbol files which exist only in other projects. This is done using a file called M2Path, which contains the names of other projects to be searched for symbol files. Each project can have such a file, and there is a global one in the S: directory. The search proceeds as follows. The compiler first looks for a symbol file in the current project's SYM directory. If the symbol file isn't there, the compiler searches through all the projects listed in the current project's M2Path file. If the symbol file is still not found, the compiler searches through all the projects listed in the global M2Path file.

This setup works nicely from the Workbench environment. The editor and compiler have attractive icons, and both the editor and the linker can generate icons for the files they produce. All you have to do to compile a program is click on the program's icon and then double-click on the compiler's icon while holding down the SHIFT key. The same process goes for linking.

Because I'm an old fashioned kind of guy, I prefer using the compiler and linker from the CLI, which gives me greater control over the various options that can be set at invocation time.

The Compiler

The compiler is one-pass, which makes compilation quite fast, compared to multi-pass compilers. When invoked from the Workbench, the compiler opens its own output window in which it shows the progress of compilation. If the compiler is launched specifically to compile one file, then it exits when the compilation is finished. However, if the compiler is launched without specifying any files to compile, then the compiler prompts the user for the name of the input file, compiles it, and then prompts for input again. It will keep going through this cycle until a blank line is entered for the input file name.

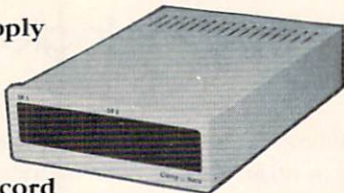
I came across an undocumented "feature" of the compiler: If the original CLI's window (the one in which LoadWB was executed) is still around when the compiler is invoked from the Workbench, then the compiler's output goes to the old

(continued)

AMIGA DUAL 3 1/2" DISK DRIVES

100% Compatible with
Amiga 500, 1000 & 2000 Computers

- Internal Power Supply
- All Metal Chassis
- Horizontal Layout
- Vented Enclosure
- On-Off Switch
- 6 foot 3-prong linecord
- Primary Circuit Breaker Protected
- Color Coordinated to Amiga Computers



ONLY \$395.00

20 Meg Hard Drive (SCSI) with Controller

ONLY \$785.00

Over 400 PD Disks. Call for complete catalog

Comp-U-Save

414 Maple Avenue, Westbury, NY 11590

In NY State (516) 997-6707

Outside NY State (800) 356-9997

"Meeting the Needs of People in the Electronic Age"

CLI window even though the compiler opened one itself. Attempting to close the compiler's window might crash the Amiga.

Launched from the CLI, the compiler behaves in the same way as when it's launched from the Workbench, except it does not open its own window. Instead, it uses the CLI's window. One annoying feature is that, for some strange reason, the compiler changes the window's attribute from CON: to RAW:, which prevents the use of ConMan, a utility I've grown to depend on. The problem is exacerbated by the fact that, unlike the other Amiga Modula-2 compilers, this compiler requires the full name of the file to be entered. I have been informed that this "feature" (messing around with the window) will be fixed in the future.

Another problem is that the compiler can be exited too easily; simply entering a blank line is enough to exit the compiler. Since the compiler is quite large and cannot be made resident, it must load from disk every time it runs. This is a lengthy and annoying process. Using FACC seems to help the load time a bit, but it would have been nice to make the compiler resident or make it more difficult to exit.

There are some documented bugs and limitations, most of which I either did not encounter, or didn't consider to be serious enough to mention. For example, the size of open

array parameters must be some power of 2. This is quite bizarre, but it might be fixed in the next version. A more common limitation is the data size limit of 32K per module. Both the Benchmark compiler and the first release of the TDI package had this limitation.

The Linker

The linker's behavior, when executed from the Workbench or from CLI, parallels the compiler's. Here, too, the full file name must be entered. For example, if you want to link a program called foo, you must enter FOO.OBJ. I find this very silly, since the compiler always generates a object file with the .OBJ suffix; why couldn't the linker implicitly append a .OBJ to the supplied file name?

The Manual

The Modula-2 manual is about the size of the TDI manual. It has one chapter about the editor, one about the compiler and language implementation, one discussing the linker, and a section listing all the support definition modules. The documentation for the compiler and editor is barely adequate. Fortunately, the authors realized this and included a reasonably good book on the topic. However, there are plenty of implementation-specific items which should have been discussed much more thoroughly in the manual (the run-time support module, for example).

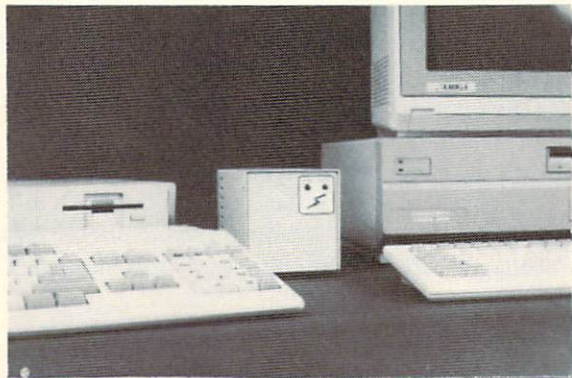
The style of the definition module listings is atrocious! I cannot believe these were meant to be read. Conventional indentation is nonexistent and the text is cramped, with practically no white space at all. Reading this section is a miserable experience. The package I received contained no index or cross reference for the support modules. This is totally unacceptable; without such a cross reference, any serious programming will become tedious or even impossible, since the programmer must scan the support module listings manually for all Amiga-related variables and types. I have been promised that this will be rectified immediately, and that future M2Amiga packages will contain a cross reference.

The Editor

The editor is based on the Emacs, supplied by Commodore on one of its developer's disks. I feel that the authors would have done better to go with a variation of mg (formerly known as MicroGnuEmacs) instead. Mg is a more recent and advanced version of Emacs. The Benchmark package uses such an adaptation of mg to its advantage.

M2Amiga's Emacs knows about compilation errors and can list them one at a time while placing the cursor on top of the offending code. This is the only Modula-2 related

The All New SS-20 Fixed Drive System for Amiga 500, 1000 and 2000 Computers



The Model SS-20 is fully compatible with all three AMIGA Computers: Model 500, 1000, and 2000. It is a 3.5 inch MiniScribe winchester drive with SCSI interface housed in a small enclosure. The SS-20 comes complete and ready for use: no additional hardware or software is necessary for operation. The back of the SS-20 chassis is fitted with a SCSI port connector so that as many as six additional SCSI devices may be "daisy-chained" to the SS-20. The Amiga's parallel port provides the interface to the SS-20. An extension of the parallel port is brought out to the rear of the SS-20 chassis for simultaneous use by other peripherals.

NEW FAST VERSION 2.4 SOFTWARE DRIVER*

The new SS-20 runs under Amiga Dos 1.2 or later. Easy to use startup utility to install the drive as a DOS device. Diagnostic utilities included to verify and test for correct operation of the unit. The driver installs during startup sequence of Workbench to appear as a drive icon. The new version 2.4 Software is compatible with Amiga "Fast File System" and future releases.

To place orders or request additional information contact:

Epic Sales Inc., Garland, Texas, (214) 272-5724

Intercomputing, Inc., Grand Prairie, Texas, (214) 988-3500

**The software is licensed from Micro Botics Inc.*



501 Business Parkway
Richardson, TX 75081
(214) 680-8394

DEALER INQUIRIES INVITED

feature of this editor worth mentioning. It is certainly not as good as the Benchmark implementation, but it is superior to that supplied by TDI.

Support Modules

M2Amiga's authors took a completely different approach from the Benchmark and TDI implementors. Instead of following the examples set by the C include files, M2Amiga's creators decided to merge most of the support stuff into a few larger modules.

The naming convention for Identifier is also quite different: types and variables begin with a lower case, while procedures begin with upper case. So, for example, a field which is named TopEdge in both the TDI and Benchmark implementation would be named topEdge in M2Amiga. The authors claim this naming convention is used by Niklaus Wirth and other Modula-2 programmers in Europe, but I find it strange.

This naming convention does not pose a serious problem if you are writing Modula-2 code for the first time. Of course, if you intend to port some code from one of the two other Amiga Modula-2 compilers, you will be faced with the

tedious task of manually converting the case of all types and variables, besides having to find the correct module from which these identifiers are to be imported.

Language Definition Issues

The M2Amiga package differs from the other two Modula-2 implementations on what might be called philosophical issues. I'll just mention one issue, which I think is a major departure from current Modula-2 thinking. Until now, Modula-2 supported the concept of "type transfer" as long as the types involved were of the same size (the equivalent to typecasting in C). For example, suppose you have the following code fragment:

```
VAR b : BITSET; c : CARDINAL; r : REAL;
b := BITSET(0);
c := CARDINAL(b);
WriteCard(c,1);
```

In the above code, the value of b is placed into c without any conversion or processing. What actually gets printed, by the way, is "1" since bit 0 was turned on. Doing something like r := REAL(c) is nonsensical, since REALs have a special format and, without conversion, the value in

(continued)



Robot Readers a powerful new way for your child to learn to read

Even if your child isn't a reader yet he can read these classic stories at his own speed through interactive speech, with little or no adult supervision. The beautiful illustrations and built-in word games hold the young reader's attention while promoting early reading skills, vocabulary, and word recognition.

*CHICKEN LITTLE *AESOP'S FABLES
*LITTLE RED HEN *THREE LITTLE PIGS

\$29.95 each
for the Amiga 512k

Coming soon: * THE UGLY DUCKLING

HILTON ANDROID
PO Box 7437 • Huntington Beach, CA 92615-7437
(714) 963-4584

c is meaningless as a REAL. In M2Amiga, however, r := REAL(c) does make sense, since the compiler will do the necessary conversion. As a matter of fact, there is no automatic type transfer anymore; instead, there's automatic type conversion. If type transfer is truly needed, a special procedure called Cast must be used. The above code will not compile under M2Amiga, but it works fine when done by the other two compilers.

This type conversion feature is quite foreign to current Modula-2 programmers, and it will take a while to get used to. It is also bound to create problems for anyone attempting to port code from other Modula-2 implementations.

M2Amiga's Forte

The M2Amiga package really shines in handling runtime exceptions and termination. The runtime support module lets a programmer designate a terminating procedure for each programmer-supplied module. The module initialization feature has always been standard in Modula-2 implementations, but a similar feature for terminating code has been sorely lacking. This new innovation is certainly welcome. Upon exiting, the runtime system calls all the programmer-designated terminating procedures, closes all opened libraries, releases the memory allocations, and closes

all windows which were opened through its Windows module. This allows the user to release all allocated resources in an orderly fashion.

If a runtime error is encountered, the runtime system displays a requester to notify the user of the error, and then attempts to shut down everything gracefully. The programmer has the option of supplying a debug procedure which gets called when a runtime error occurs. This means it would be quite simple to incorporate a source level debugger and, as a matter of fact, I know that such a debugger is in the works. A source level debugger greatly eases debugging, thereby speeding development time.

Comparison

M2Amiga will compete against the Benchmark package. Both are one-pass compilers, and both use some version of Emacs as editor. They are also within the same price range (M2Amiga lists for about \$250). Execution time for the executables produced by the two compilers is pretty much the same, but the size of the M2Amiga generated executable is larger because its runtime module has more overhead to take care of.

The M2Amiga compiler is slower than Benchmark, especially when importing certain large modules (such as Intuition), but it is still much faster than the TDI implementation. Look at the accompanying tables for some Benchmark results.

Although M2Amiga's user interface is better than TDI's, I found it is much worse than the Benchmark implementation. Frankly, I've been spoiled by the well-integrated Benchmark environment, so anything else pales in comparison.

The M2Amiga manual and support module listings are probably the worst of all the Modula-2 packages, but that should change in the future.

Summary

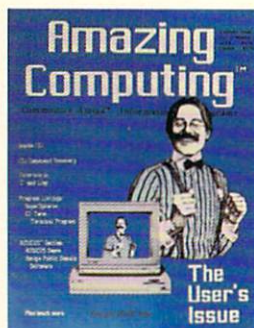
The Interface Technologies M2Amiga Modula-2 package is a viable alternative in the Amiga compiler field. Compared to the Benchmark package, M2Amiga has both strengths and weaknesses; individual programmers must decide which package best suits their needs. One factor to keep in mind is that Interface Technologies seems more stable than Benchmark's distributors; M2Amiga's future might therefore be more sound.

M2Amiga Interface Technologies

3336 Richmond, Suite 200
Houston, TX 77098
Tel: (713) 523-8422

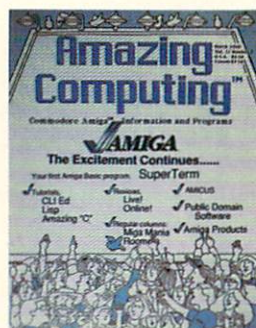
(continued on page 106)

Expanding Reference



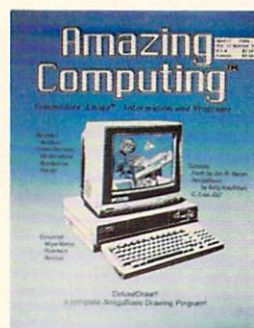
Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kaufman An ABASIC Graphics prog.
Date Virus By J. Foust A disease may attack your Amiga!
EZ-Term by Kelly Kaufman An ABASIC Terminal program
Mega Menu by P. Kivowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDOS™
CLI Summary by G. Musser A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products



Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kaufman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Winch
Amiga BBS numbers



Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viveiros
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw! by R. Winch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE



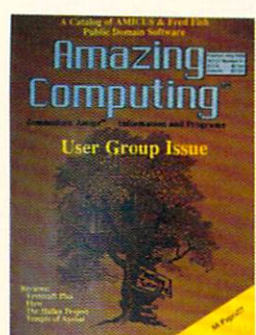
Volume 1 Number 4 May 1986

SkyFox and Artiofox Reviewed
Build your own 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rick Winch
Scrimper Part One by P. Kivowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keefe
Amiga BBS Numbers



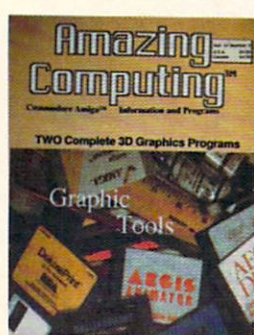
Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
 by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidcar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidcar affect the Transformer
 an interview with Douglas Wyman of Smile
The Commodore Layouts by J. Foust A look Commodore "outs"
Scrimper Part Two by Perry Kivowitz
Marsuder reviewed by Rick Winch
Building Tools by Daniel Kary



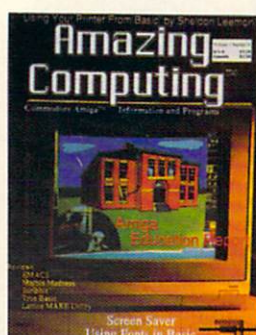
Volume 1 Number 6 1986

Temple of Apokal Trilogy reviewed by Stephen Pietrowicz
The Halley Project: A Mission in our Solar System
 reviewed by Stephen Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kaufman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz



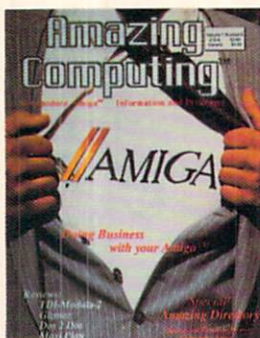
Volume 1 Number 7 1986

Angle Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Amiga Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Str! by B. Casey program Amiga Basic menus
IFF Brush to AmigaBasic "BOB" Basic editor by M. Swinger
Linking C Programs with Assembler Routines on the Amiga
 by Gerald Hull



Volume 1 Number 8 1986

The University Amiga By G. Garble
 Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scrimper Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & The Talking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen Saver by P. Kivowitz A monitor protection prog. in C
Latex MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmp File Reader in Amiga Basic by T. Jones



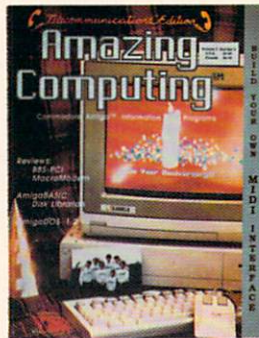
Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegre Memory Board Reviewed by Rich Wirth
TxEt Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amiga and Fred Fish PDS
Doe 2 Doe review R. Knepper
Transfer files from PCMS-DOS and AmigaBasic
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmox reviewed by Peter Wayner Amiga extra!
The Loan Information Program by Brian Catley
basic prog. for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absorb Amiga Fortran Compiler reviewed by R. A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
58000 Macro on the Amiga by G. Hull Advance your ability.
TDI Modula-2 Amiga Compiler review by S. Fawcett



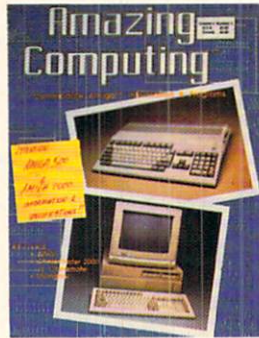
Volume 2 Number 1 1987

What Digi-View Is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley
A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell
Using Lattice C with a single drive system
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint L... by New Tek reviewed by John Foust
Deluxe Paint II... from Electronic Arts reviewed by J. Foust



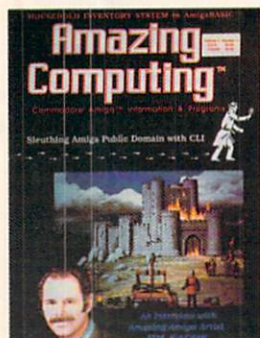
Volume 2 Number 2 1987

The Modern by Joseph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project... Graphic Teleconferencing on the Amiga
by S. R. Pietrowicz
Flight Simulator II... A Cross Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C



Volume 2 Number 3

The Amiga 2000™ by J. Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amiga
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shorts
Intuition Gadgets by Harriet Maybeck Tolley
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewed by Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercoff
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?"
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo*



Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Smashing Public Domain Disks with CLI by John Foust
Highlights from the San Francisco Commodore Show
by S. Hull
Speaker Sessions: San Francisco Commodore Show H Toly
The Household Inventory System in AmigaBASIC™
by B. Catley
Secrets of Screen Dumps by Nadun Okun
Using Function Keys with MicroEmacs by Greg Douglas
AmigaTrix II by Warren Block More Amiga shorts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti's Pele football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amiga Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Toly An easy-to-use debugger



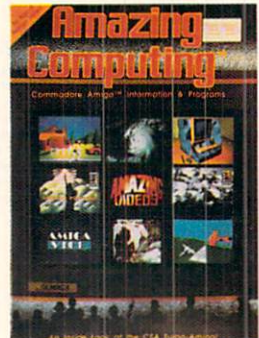
Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryan comparing JForth and Muf-Forth
Basic Input by B. Catley AmigaBASIC input routine for use in
all your programs.
Writing a SoundScape Module in C by T. Fay Programming
with MIDI, Amiga and SoundScape by SoundScape author.
Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real digitized STEREO
AmigaNotes by R. Rae A review of Mimetix
SoundScape Sound Sampler.
More AmigaNotes by R. Rae
A further review of Sunrise's Perfect Sound.
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
The Mimetix Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetix' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.



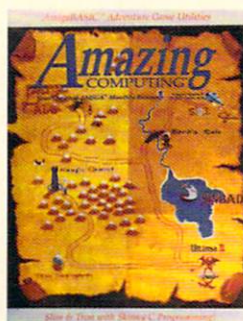
Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM kernel.
The Amazing Computing Hard Disk Review by J. Foust & S.
Loeman In-depth looks at the C Ltd. Hard Drive, Microbotics'
MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive
and Xebec's 9720H Hard Drive. Also, a look at disk driver
software currently under development.
Module-2 AmigaDOS™ Utilities by S. Fawcett A
Calls to AmigaDOS and the ROM kernel.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amiga Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
High Level Shakeup Replaces Top Management at
Commodore by S. Hull
Peter J. Baczorby S. Hull Manager at CBM gives an inside look
Logixix A review by Richard Knepper
Organizer by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H. Maybeck Toly



Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amiga & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust.
Amazing Stories and the Amiga™ by John Foust.
All about Printer Drivers by Richard Blak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik
Easy! drawing tablet by John Foust.
CSA's Turbo-Amiga Tower by Alfred Abuto
68000 Assembly Language by Chris Martin.



Volume 2, Number 8 1987

This month's Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

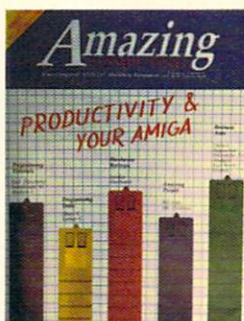
SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I, II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bird's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
Disk-2-Disk by Matthew Leeds
The ColorFont Standard by John Foust
Skinny C Programs by Robert Remersma, Jr.
Hidden Messages in Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex by J. Foust



Volume 2, Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepeli
Life, Part I: The Beginning by Gerald Hill
The ultra-complex nine bit solution to the "Game of Life," Amiga Visual by John Foust
A new Amiga virus has surfaced. Please check your system.
CLI Arguments in C by Paul Castonguay
MIDI Interface Adapter by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
Module-2 by Steve Fawcett
First in a series, a command line calculator in Module-2.
AmigaNotes by Richard Rae
The audio changes made in the Amiga 500 and 2000.
Animation for C Rookies: Part III by M. Swinger
tackling double-buffering.
The Big Picture by Warren Ring
Amiga™ Assembly language programming for the brave!
Karate Kid Review by Stephen R. Pietrowicz
GO! 54 review by John Foust, James O'Keane, and Rick Wich
Three C-64 experts investigate a new Amiga 54 emulator.
A-Talk-Plus Review by Brendan Larson
"Full-fledged terminal program" & Tektronics capabilities
Calligrapher Review by John Foust
Animator: Apprentice Review by John Foust
Playing Dynamic Drums on the Amiga by David N. Blank
WordPerfect Review by Steve Hull
Insider/Kwikart Review by Ernest P. Viveiros Sr.
RAM & ROM expansion: Comments and installation tips.
Bug Bytes by John Steiner
Forth! by Jon Bryan
DumpPort utility for your Multi-Forth toolbox.
As I See It by Eddie Churchill
An offset look on Digi-Paint, Portal, and Videospace 3D.
The Amicus Network by John Foust
The Commodore Show and AmExpo: New York!



Volume 2, Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Wich
Gizmox Productivity Set 2.0 review by Bob Eller
Kizwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Fawcett
Leather Goddess of Phobos
reviewed by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.1a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leeman
AC-BASIC Compiler an alternative comparison by B. Catley
Module-2 Programming S. Fawcett
Raw Console Device Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscapes
Todor Fay manipulate's samples
Bill Volk, Vice-President Agile Development,
interviewed by Steve Hull
Jim Goodnow, Developer of Manx 'C'
interview by Harriet M. Tolly



Volume 2, Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing
Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
Bug Bytes by John Steiner
Amiga Notes by Richard Rae
Roomers by The Bandito
68000 Assembly Language by Chris Martin
The Amicus Network by John Foust
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Module-2 by Steve Fawcett
Window I/O by Read Predmore



Volume 2, Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble, and WordPerfect compared
LPD Writer Review by Marion Deland
ViziPaint Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Berovitz
The author of StarGlider speaks!
Do-It-Yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: a quick look at an
intense grammar examination application
As I See It by Eddie Churchill. Peeking at WordPerfect,
Gizmox V2.0 and Zing! Keys
Bug Bytes by John Steiner
AmigaNotes by R. Rae 4 electronic music books
Module-2 Programming by Steve Fawcett
devices, I/O, and the serial port
Roomers by The Bandito
68000 Assembly Language by Chris Martin
Chris walks through the display routines
The Amicus Network by John Foust
Desktop Publishing & Seybold
C Animation Part II by Mike Swinger Animation Objects
BASIC Text by Brian Catley Pixel perfect text positioning
Soundscapes Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barnes
File Browser by Bryan Catley
Full Feature BASIC File Browsing utility

Your Resource to the Commodore Amiga™

The phrase above is not just empty words. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Back Issues contains articles from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, indepth reviews and hands on articles for their machines.

Amazing Computing™ was the first magazine to document CLI
Amazing Computing™ was the first to show Sidecar™ from COMDEX™ in full detail.
Amazing Computing™ was the first to document a 5 1/4 drive connector
Amazing Computing™ was the first with a 1 Meg Amiga upgrade hardware project!
Amazing Computing™ was the first magazine to offer serious programming examples and help.
Amazing Computing™ was the first to offer Public Domain Software at reasonable prices.
Amazing Computing™ was the first magazine with the user in mind!

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amiga. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga that any user will find useful.

\$4.00 each!

Our Back Issue price is still \$4.00 per issue! (Foreign orders, please add \$1.00 per issue for Postage & Handling. All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.)

Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Please complete your Amazing Computing™ library today, while these issues are still available. Please complete the order form in the rear of this issue and mail with check or money order to:

Back Issues, PIM Publications, Inc., P.O. Box 869, Fall River, MA 02722
(Please allow 4 to 6 weeks for delivery)

5 Reasons Why You're Ready For MacroModem

1. You love telecom, but not memorization. MacroModem's user-written macro libraries and companion help screens (36 macros per file) store log on procedures, remote system menus and commands, ...
2. You've always wanted to use the mouse after you're connected, too. Write macros that mimic remote system commands and menus, then execute them with the mouse or keyboard.
3. You like automation, but not script languages. Our macros use normal commands from MacroModem, remote systems, and AmigaDOS, as well as text and control codes. A multi-windowed MacroEditor is included. No new programming language to learn.
4. You want to do other things while downloading a file. MacroModem is truly multi-tasking, with a NewCLI available anytime, even during file transfers. And MacroModem's error checking won't stop downloads unless you tell it to.
5. Of course MacroModem includes standard telecom software features, too. Teach MacroModem what you want, and it will remember for you.

MacroModem - the better way to do telecommunications. \$69.95

Kent Engineering & Design
P.O. Box 178, Mottville, NY 13119
(315) 685-8237

Benchmark Results

1. Please refer to the Vol. 1.9 issue of Amazing Computing for the listing of the benchmark programs.
2. All compiles were done from VD0: to VD0:. All links were done off VD0:.
3. All runtime checks were disabled.
4. TDI note: The compiler was run resident. The optimize option was at the link stage.
5. Benchmark note: The compiler was loaded into the editor and executed from there. For comparison's sake, I've used the InOut module in all programs, even though the compiler comes with a module called TermInOut which is only slightly more limited; but it can cut down the executable size by 50%.
6. M2Amiga note: The compiler and linker were loaded into separate CLIs and never exited, to avoid having them reload for disk every time.

Primes

	TDI	M2Amiga	Benchmark
Compile Time	3	1.8	0.9
Link Time	13.5	1.5	2.2
Build Time	16.5	3.3	3.1
Run Time	53	51	51
Obj (.Lnk) Size	344	460	196
Executable Size	4692	11980	7656

Sieve

	TDI	M2Amiga	Benchmark
Compile Time	4	2	1
Link Time	14	3	3
Build Time	18	5	4
Run Time	6.2	5.8	5.9
Obj (.Lnk) Size	734	668	384
Executable Size	5024	12188	16064

Window

	TDI	M2Amiga	Benchmark
Compile Time	38	12	4
Link Time	25	3	2.9
Build Time	63	15	6.9
Obj (.Lnk) Size	1368	840	343
Executable Size	4512	12136	9828

{ Continued CALC Listings }

```
IMPLEMENTATION MODULE MyRealOut;
(* ..... *)
(*      (c) Copyright 1986, 1987      *)
(*      by Steve Falwiczewski        *)
(* ..... *)
(*      For non-commercial, non-profit use only *)
(* ..... *)

FROM InOut IMPORT WriteString, Write;
FROM Conversions IMPORT ConvertToString;

PROCEDURE Conv(u: LONGCARD);
VAR
  line: ARRAY[0..79] OF CHAR;
  done: BOOLEAN;
BEGIN
  ConvertToString(u, 10, FALSE, line, done);
```



```

WriteString(line)
END Conv;

PROCEDURE RealToLongCard(VAR r: REAL; VAR cu: CARDINAL);
(* This is a dirty hack, but it works! *)
VAR
  overflow: BOOLEAN;
  u: LONGCARD;
BEGIN
  overflow := FALSE;
  IF r > 65535.0 THEN
    overflow := TRUE;
    r := r - 65535.0;
  END;
  cu := TRUNC(r);
  u := LONGCARD(cu);
  IF overflow THEN
    u := u + 65535;
  END;
  Conv(u);
END RealToLongCard;

PROCEDURE WriteReal(r: REAL; decimal: CARDINAL);
VAR
  f, u: LONGCARD;
  i, cu: CARDINAL;
BEGIN
  IF r < 0.0 THEN
    Write('-');
    r := -1.0 * r;
  END;
  RealToLongCard(r, cu);
  Write('.');
  FOR i := 1 TO decimal DO
    r := r - FLOAT(cu);
    r := r * 10.0;
    cu := TRUNC(r);
    Write(CHR(cu + ORD('0')));
  END;
END WriteReal;

END MyRealOut.

IMPLEMENTATION MODULE MyInOut;
FROM InOut IMPORT Read;
CONST LF = 12C;
PROCEDURE ReadString(VAR s: ARRAY OF CHAR);
VAR
  i: CARDINAL;
BEGIN
  i := 0;
  REPEAT
    Read(s[i]);
    INC(i);
  UNTIL (i > HIGH(s)) OR (s[i-1] = LF);
  IF s[i-1] = LF THEN s[i-1] := 0C END;
END ReadString;
BEGIN
  END MyInOut.

IMPLEMENTATION MODULE Interpreter0;
(* ***** *)
(* (c) Copyright 1986, 1987 by *)
(* Steve Faiwischewski & Richie Bielak *)
(* For non-commercial, non-profit use only. *)
(* ***** *)
(*SQ*) (* Change all JSRs to BSRs in this module *)

FROM InOut IMPORT WriteString, WriteCard, WriteLn;
FROM Strings IMPORT Length;
FROM Storage IMPORT DEALLOCATE;

PROCEDURE WriteLine(VAR line: ARRAY OF CHAR);
BEGIN
  WriteString(line);
  WriteLn;
END WriteLine;

PROCEDURE Trim(VAR sss: ARRAY OF CHAR);
(* Trim trailing blanks *)

```

DYNAMIC DRUMS

The program that transforms your Amiga™ into a professional drum machine.

- Incredibly realistic sound
- Create your own studio-quality drum tracks
- Real or step time programming
- Graphic Editing
- Over 100 percussion samples included or use your own IFF samples
- Fully adjustable volume and tuning levels
- Randomizing options for a dynamic, human feel
- MIDI compatible

Requires 512K Amiga™
MI, FL, & CA add sales tax

DEALER INQUIRIES INVITED

Send Check or Money Order for \$79.95 to:

NEW WAVE
S O F T W A R E

P.O. Box 438, St. Clair Shores, Michigan 48080

(313) 771-4465

Amiga is a trademark of Commodore-Amiga Inc.

```

VAR
  i: INTEGER; (* not CARDINAL because can go negative *)
  doit: BOOLEAN;
BEGIN
  i := Length(sss) - 1;
  doit := TRUE;
  WHILE doit AND (i >= 0) DO
    IF sss[i] = ' ' THEN
      sss[i] := 0C;
      DEC(i);
    ELSE
      doit := FALSE;
    END;
  END;
END;
END Trim;

PROCEDURE FreeNodes(VAR t: nodep);
(* ***** *)
(* We dispose of the binary tree pointed to by t in *)
(* a recursive manner. We first free up the left son *)
(* (if there is one), and then we do the same to the *)
(* right son. Then we dispose of t, and set it to NIL *)
(* ***** *)
BEGIN
  IF t <> NIL THEN
    IF t^.lson <> NIL THEN FreeNodes(t^.lson) END;
    IF t^.rson <> NIL THEN FreeNodes(t^.rson) END;
    DISPOSE(t);
    t := NIL;
  END;
END FreeNodes;

END Interpreter0.

```

(continued)

Special Holiday Price....

KLINE-TRONICS'
1 MEG Ram Expansion
AMMEG1™ only \$249.95*

To thank AMIGA™ owners everywhere for the great response to our ads, Kline-Tronics is offering this special Holiday price. This price is only available directly from Kline-Tronics. Order now before the rush. **Regularly \$299.95**

- 1 Meg* "FAST" Ram in Metal Case
 - True "Auto-Configure"
 - Fully Assembled & Tested
 - 90-Day Parts & Labor Warranty
- (* All Ram Chips Included)

"HIGH QUALITY" at a "LOW PRICE"

KLINE-TRONICS

10 Carlisle Court York, PA 17404
 Tel. (717)-764-4205

* Plus Shipping & Handling
 Limited Time Offer

IMPLEMENTATION MODULE Interpreter:

```
(*****
(*)
(*) (c) Copyright 1986, 1987 by
(*) Steve Faliszewski & Richie Bielak
(*)
(*) For non-commercial, non-profit use only.
(*)
(*****)
```

```
($Q+*) (* Change all JSRs to BSRs in this module *)
($S-*)
($T-*)
```

```
FROM InOut      IMPORT WriteString, WriteLn, Write,
                  WriteCard, WriteInt;
FROM MathLib0    IMPORT sin, cos, tan, log, ln, exp, sqrt,
                  real, pi, entier;
FROM Storage     IMPORT ALLOCATE, CreateHeap, DestroyHeap,
                  HeapLeft;
FROM Strings     IMPORT CompareResults, Compare, Length,
                  InitStringModule;
FROM CommandLine IMPORT CLStrings;
FROM Interpreter0 IMPORT Trim, WriteLine, FreeNodes, nodep,
                  maxtree, maxoper, treestack,
                  operstack;
```

```
CONST
  CmdSet = CharSet('S','C','T','L','E','A');
  DigitSet = CharSet('0'..'9');
  OperSet = CharSet('^','+', '-', '*', '/', ' ');
```

```
VAR
  t1 : nodep; (* hide this *)
  linesize : CARDINAL;
```

```
PROCEDURE ItsAFunction(oper : CHAR) : BOOLEAN;
BEGIN
  RETURN (oper IN (CmdSet + CharSet('Q','N')));
END ItsAFunction;
```

```
PROCEDURE precedence(operator : CHAR) : INTEGER;
VAR p : INTEGER;
```

```
BEGIN (* precedence *)
  IF operator IN (OperSet + CharSet('^',' ','/','*')) THEN
    CASE operator OF
      '^','/' : p:=0;
      '+','-' : p:=1;
      '*', '/' : p:=2;
      '^' : p:=4;
      '/' : p:=1;
    ELSE
      WriteLn('Precedence: Program Error!!!');
      p := -2;
    END
  ELSE
    WriteLn('Precedence: Program Error!!!');
    p := -2;
  END
  ELSIF ItsAFunction(operator) THEN
    p := 8;
  END;
  RETURN(p);
END precedence;
```

```
PROCEDURE nexttoken(VAR line: CLStrings;
  VAR LineIndex: CARDINAL) : nodep;
```

```
VAR
  i : CARDINAL;
  fun : ARRAY[0..2] OF CHAR;
  m,
  sum : REAL;
  t1 : nodep;
  done: BOOLEAN;
```

```
PROCEDURE CheckForFuncs(VAR line: CLStrings; VAR t1: nodep);
VAR
```

```
  i : CARDINAL;
BEGIN
  IF line[LineIndex+1] IN
    CharSet('B','X','O','I','A','Q','N') THEN
    what:=func;
    FOR i:=0 TO 2 DO
      fun[i]:=line[LineIndex+i];
    END;
    IF Compare(fun,'SIN')=Equal THEN t1^.oper:='S'
    ELSIF Compare(fun,'COS')=Equal THEN t1^.oper:='C'
    ELSIF Compare(fun,'TAN')=Equal THEN t1^.oper:='T'
    ELSIF Compare(fun,'LOG')=Equal THEN t1^.oper:='L'
    ELSIF Compare(fun,'EXP')=Equal THEN t1^.oper:='E'
    ELSIF Compare(fun,'ABS')=Equal THEN t1^.oper:='A'
    ELSIF fun[0]='L' THEN t1^.oper:='N'
    ELSIF Compare(fun,'SQR')=Equal THEN
      t1^.oper:='Q';
      IF line[LineIndex+3]='T' THEN
        INC(LineIndex);
      END;
    END;
    ELSE what:=oops; END;
  IF what=func THEN
    IF t1^.oper = 'N' THEN
      INC(LineIndex,2)
    ELSE
      INC(LineIndex,3)
    END;
    done:=TRUE;
  END;
END CheckForFuncs;
```

```
PROCEDURE CheckForVariables;
```

```
VAR
  i : CARDINAL;
BEGIN
  i := ORD(line[LineIndex]) - ORD('A');
  IF valdef[i] THEN
```

```
(* Ok, the variable is defined, but instead of copying *)
(* its value, lets just keep a reference to it. This is *)
(* usefull if the variable's value may be changed after *)
(* the tree is built. *)
```

```
  t1^.val := FLOAT(i);
  t1^.oper := 'V';
  what := number;
  INC(LineIndex);
  done:=TRUE;
ELSE
  HandleError(MissingValueError,ORD(line[LineIndex]));
  what := oops;
END;
END CheckForVariables;
```

```
PROCEDURE CheckForNumbers;
```

```
BEGIN
  what := number;
  sum := 0.0;
  WHILE (line[LineIndex] IN DigitSet) AND
    (LineIndex<=linesize-1) DO
    sum := sum * 10.0 + real(ORD(line[LineIndex]) -
      ORD('0'));
    INC(LineIndex);
  END;
END;
```



```

        sign := sign * (-1)
    ELSE
        HandleError(MissingOperandError,2);
        what := oops;
    END
END
ELSE
    (* if precedence of the operator on the top *)
    (* of the operator stack is greater than or *)
    (* equal to the precedence of the scanned *)
    (* operator then pop the operator stack and *)
    (* build subtrees. *)
*)
(* We assign the precedence values to variables and *)
(* compare the variables, instead of simply comparing *)
(* the function itself (i.e. do *)
(* IF precedence(t1^.oper) <= *)
(* precedence(operstack[ostack]^.oper) *)
*)
*)
(* because of a bug in version 2 of the TDI package. *)

    p1 := precedence(t1^.oper);
    p2 := precedence(operstack[ostack]^.oper);
    IF p1 <= p2 THEN
        PushPopGrow;
    END;
    whatwas := operator;
    INC(ostack);
    operstack[ostack] := t1
END
END;
END ProcessOperator;

PROCEDURE ProcessEndOfLine(VAR sign: INTEGER);
BEGIN
    (* clean up the operator the stacks *)
    (* if there is a '(' on the stack a ')' was missing *)
    WHILE ostack > 1 DO
        IF operstack[ostack]^.oper = '(' THEN
            ostack := 1; (* i.e. giveup *)
            HandleError(MissingCloseParenError,0);
            what := oops;
        ELSE
            (* if the tree stack under flows *)
            (* then we're short an operand *)
            IF tstack <= 1 THEN
                ostack := 1;
                HandleError(MissingOperandError,3);
                what := oops;
            ELSE (* its ok *)
                PushPopGrow;
            END;
        END;
    END;
END ProcessEndOfLine;

PROCEDURE GrowTree(VAR line: CLStrings;
    VAR LineIndex: CARDINAL);
    (***** *)
    (* Build the binary evaluation tree *)
    (***** *)
VAR
    sign : INTEGER;
    t1 : nodep;
BEGIN (* grow tree *)
    FreeNodes(treestack[1]);
    linesize:= Length(line);
    sign := 1;
    REPEAT
        t1 := nexttoken(line,LineIndex);
        CASE what OF
            (* A number was scanned, just push it *)
            (* onto the tree stack. *)
            number: (* A number cannot follow a number *)
                IF whatwas = number THEN
                    HandleError(MissingOperatorError,0);
                    what:=oops;
                ELSE
                    INC(tstack);
                    t1^.val := t1^.val * real(sign);
                    sign := 1;
                    treestack[tstack] := t1;
                    whatwas := number
                END
        END
    UNTIL
        func:
            IF whatwas<>operator THEN
                HandleError(MissingOperatorError,0);
                what:=oops;
            ELSE
                INC(ostack);

```

```

        operstack[ostack]:=t1;
        whatwas:=func;
    END

    operator:
        ProcessOperator(sign,t1)
    endofline:
        ProcessEndOfLine(sign)
    oops:
        HandleError(GeneralError,0);
    ELSE
        WriteLine('** GrowTree: Program Error **');
        END (* CASE *)
        UNTIL what IN tokenset{oops,endofline};
    END GrowTree;

PROCEDURE SubstituteVar(root : nodep): REAL;
VAR
    sign : REAL;
BEGIN
    sign := 1.0;
    WITH root^ DO
        IF val<0.0 THEN sign := -1.0; END;
        (* val*sign gives ABS(val) *)
        RETURN(sign*symbtable[TRUNC(val*sign)]);
    END;
END SubstituteVar;

PROCEDURE Evaluate (root : nodep; VAR suc: BOOLEAN) : REAL;
CONST
    MaxExp = 11.6;
VAR
    i : CARDINAL;
    x,p : REAL;
BEGIN
    IF (root<>NIL) AND suc THEN
        CASE root^.oper OF
            '+' : RETURN(Evaluate(root^.lson,suc) +
                Evaluate(root^.rson,suc))
            '-' : RETURN(Evaluate(root^.lson,suc) -
                Evaluate(root^.rson,suc))
            '*' : RETURN(Evaluate(root^.lson,suc) *
                Evaluate(root^.rson,suc))
            '/' :
                x := Evaluate(root^.rson,suc);
                IF x <> 0.0 THEN
                    RETURN(Evaluate(root^.lson,suc) / x)
                ELSE
                    HandleError(DivideByZeroError,0);
                    what:=oops;
                END
            'A' : RETURN(ABS(Evaluate(root^.lson,suc)))
            'S' : RETURN(sin(Evaluate(root^.lson,suc) *
                pi/180.0))
            'C' : RETURN(cos(Evaluate(root^.lson,suc) *
                pi/180.0))
            'T' :
                x:=Evaluate(root^.lson,suc);
                p:=cos(x*pi/180.0);
                IF p<>0.0 THEN RETURN(sin(x*pi/180.0)/p)
                ELSE
                    HandleError(IllegalValueForTanError,0);
                    what:=oops;
                END
            'E' :
                x:=Evaluate(root^.lson,suc);
                IF x < MaxExp THEN RETURN(exp(x))
                ELSE
                    HandleError(IllegalValueForExpError,0);
                    what := oops;
                END
            'L' :
                x:=Evaluate(root^.lson,suc);
                IF x>0.0 THEN RETURN(log(x))
                ELSE
                    HandleError(IllegalValueForLogError,0);
                    what:=oops;
                END
            'N' :
                x:=Evaluate(root^.lson,suc);
                IF x>0.0 THEN RETURN(ln(x))
                ELSE
                    HandleError(IllegalValueForLogError,0);
                    what:=oops;
                END
        END
    END

```

[The remainder of IMPLEMENTATION MODULE Interpreter will be presented next month - Ed]

•AC•

An Interview with Dave Haynie of Commodore/Amiga™

This is a transcript of the formal conference held in AmigaForum conference room 2 on November 11th, 1987. Our guest was Dave Haynie of Commodore/Amiga.

The following is a heavily edited transcript of that conference. Guest speakers' and moderator's comments are identified with their initials, e.g., DH. Forum members are designated with their names inside parenthesis, e.g., (Mike).

(Mike) Hi Dave, nice to see you survived the return from Vegas!

DH: Yeah, Vegas was a blast. Because everything worked. For those who weren't there, C-A really showed some new stuff this time.

RR: Dave, we're hoping you'll give us the show scoop tonight; are you willing? We'll just give you the floor and you can ramble on about COMDEX stuff in as much detail as you'd like, then we'll go for questions ... unless you have another suggestion?

DH: Okay. Starting with the reason I didn't get any sleep before Comdex, the 68020 board. This is a plug-in for your A2000; it contains a 68020, 68881 FPU, 68851 MMU, and two megabytes of RAM. You can clock the FPU as fast as you like with the addition of a clock module (and faster FPU). I expect it would work in something like the 2000-and-1 box that ASDG is planning, for all you A1000 fans.

Moving to the right, we had the PVA device. This was designed for C-A by folks at the New York Institute of Technology. It contains a high quality genlock and frame grabber in one box. Also supports two video inputs, under software control.

On the other side of this part of the booth, we had the new hi-res monitor. This gives you 1008x800x2 monochrome, and it works on any Amiga—just plugs into the video connector. Currently, you RamKick a few new libraries to get it running, but future ROM/Kickstart releases will have support built-in.

We also showed the 80286 Bridge Card. While Steve Ballmer of Microsoft calls the 80286 a "brain-damaged" chip, and I'd probably agree, for you MS-DOS fans, it's certainly faster than an 8088. 10MHz, to be exact.

Also in the booth was the A2000 internal genlock. Better quality than the 1300, and probably cheaper, too.

From 3rd parties, there was a de-interlacing device shown. Basically a scan converter that fits in your A2000 video slot and gives you up to 705x512, all 4096 colors, displayable on a multi-sync or IBM VGA monitor. If you've seen the quality of the IBM VGA, you know exactly what this looks like.

NewTek showed their AmiGen genlock device, which fits in the video slot of any Amiga. They also showed, near the end of the show, their "Video

Toaster." That's a killer machine. Does simple things like "pixelizations," or more complex things, like wrapping live video images around spheres. All the things hard-core video types AREN'T using Amiga for right now. At least, I was very impressed.

Software was shown too. Some good software. Deluxe Productions, basically a storyboarding program. Professional Page, City Desk, lots of things you may have heard about from AmiExpo. There's even more going on behind the scenes; I think the outside world is really starting to notice Amiga. My only complaint: fewer Amigas in other booths than Macintoshes, though there were some. Oh yes, except for Progressive Peripherals. They had an almost all Amiga booth. One ST, one or two C-128s, the rest Amiga. They've got a very fast frame grabber, two CAD packages, and a little program for digital signal processing and other graphic manipulations called PixMate. Does basically what the University of Lowell DSP board does, things like line detection, only not real time. Still, they use the blitter to do DSP faster than a fast '020 could do it. Anyway, I guess I can answer some questions now.

RR: Dave, before opening the floor, can you project prices or releases on any of this, or is it still too early?

DH: We didn't have any pricing available on most of the C-A products shown. They haven't been officially announced yet, though they're really

(continued)

not that far off. In the old days, these probably would have been shown in private. Still, I'm reasonably sure that the '020 with 2 meg of fast RAM will cost far less than any similar configuration for the Amiga currently on the market. Wish I had better pricing info, but it's really a marketing/sales thing to set the pricing, and they haven't been given any of our new stuff yet.

RR: Okay, fair enough. The floor is now open for questions.

(Vic) RE: Hi-Res monitor: "1008 x 800 x 2 monochrome"? What's the "2"?

DH: That's two bitplanes. Like a normal Workbench screen, except in this case it's black, white, light grey, and dark grey. You can run it in one, if you like, but for compatibility reasons, two planes is the minimum that most Workbench software will be happy with. The "1008" some of you may recognize as an Agnes number.

(Marlene/SYSOP) Dave, can you tell us more about NYIT's board, your impressions of it, and potential release date.

DH: Sure thing. As far as release, I believe it's already in FCC test, which indicates that, barring any changes to make Uncle Sam happy, the design is complete and frozen. No official word on release date, though. As for potential, I played with it some, and it's kind of neat. Really good quality genlock, on par with the best I've seen. You have an instant frame grab, though that's into local board memory; the transfer to an Amiga IFF file takes longer. It does have software which looks pretty far along too, the captured image can be converted to HAM, lores, or hires. Programs can swap video inputs, which is a feature I haven't seen yet. They had it set up on one camera and one VCR; you click on a gadget to go from VCR to camera, or from live to captured image. I'm not totally familiar with it; it may have other features as well, but what I saw of it worked very nicely.

(Paul B.) Dave, can you tell us if and when the medium/long persistence monitor will be available?

DH: Last I heard, the 2080 monitor of which you speak would be available in January, or thereabouts. I haven't personally seen one since the last demo unit at CES last January, but that's not really unusual, since the product is OEMed from outside. The one I did see was very nice, tighter dot pitch than a 1080, and with ProWrite running black on white the flicker was all but non-existent, no worse than a PAL monitor. Hope they ship as nice a unit.

(Fred Kohler) Dave, will there be any hardcards out for the Amiga 2000 that go into an Amiga slot?

DH: I haven't seen one yet. C-A isn't planning one. The decisions behind the C-A hard disk were based on realities. Like, you can get a large ST-506 drive from any PC clone garage shop vendor cheaper than you could from CBM. They'll sell you a hard drive, but integrating would kind of force you into accepting the drive WE pick for you instead of the one you really want. Maybe a 3rd party will make one.

(Dave K.) Dave, I'd like to hear more about the 68020 board. For instance, can you turn the '020's instruction cache on and off with it? Can you still use 16-bit RAM with it installed?

DH: Okay. AmigaDOS will turn the Cache on automatically on system initialization, and then leave it alone. So once you're up, you can turn it on and off at will. The on-board 32 bit RAM autoconfigures as the first 2 meg, but any extra 16 bit RAM can still be used out in the expansion bus, and 16 bit DMA devices can DMA into the on-board 32 bit RAM if they like. The 32 bit RAM runs at 14.2 MHz with one wait state (due to the MMU).

(Isaac Cruz) Good evening Dave! Any news on expanded memory (beyond 1.5M) for the A500? And how about the availability of hard disks for the same machine. Prices?

DH: Glad you mentioned that. Byte-by-Byte showed a 2 meg external, self powered memory add-on for the A500, nice unit. Detects when the A500 is powered on, sits on the left beneath an add-on floppy, if your floppy reaches over there. I think Supra already has a hard drive for the A500, and I know of at least one other hard drive for the A500, unannounced at this time, that will also probably house a meg or two of RAM as well. WARNING: DON'T BUY A NON-POWERED EXPANSION DEVICE FOR THE A500.

(Doug) A non-powered expansion device, Dave? You mean like the A1010? [Grin] My question concerns the new bridge card. The current software for the 4.17 Mhz card that's available now is pretty slow on screen scrolling, etc. Are there any new breakthroughs for changing this, since scrolling will supposedly be more intense with the new 286 bridge?

DH: Well, one A1010 is OK, after that, worry. Anyway. Yeah, I finally broke down and hooked up a bridge card myself. The color screen will be more along the lines of what you get in PC performance, I believe, since there's less Amiga-side manipulations involved. The AT card will run a bit faster, since the 80286 is running faster. But they do use the same interface chips, and the same Amiga-side software. So short of adding a 68020 card, don't expect a major speed-up until perhaps the next revision of the PC software.

(Doug) [Sigh] That's what I was afraid of. R.I.P. 286 Bridge!

RR: Dave, Steve Ahlstrom asks if you can pull the 8088 in the bridge and replace it with an NEC V20.

DH: I've never actually tried a V20, but I imagine it'll work OK. The 2088 system timing isn't extremely tight, the interface chips WERE designed to support the AT board as well. But I'm not all that familiar with the differences; trying it would be the best test, of course.

As far as RIP bridge board 286 goes, I doubt it. I never expected the 2088 to sell very well, but we can't make them fast enough, by about a factor of 3. As the PC, at least in my opinion, was never a graphics intensive machine in any stretch of the imagination, I don't think slow graphics would kill it necessarily.

(Bill Atkinson) Dave, how would you compare the MicroChannel and NuBus to the B2000's bus for performance and expansion (32 bits etc.), and, by the way, THANKS! for the B2000.

You're quite welcome! Anyway, first and foremost, the B2000 bus is very much the original Zorro bus with three interrupt lines changed somewhat. And it's very much a 16 bit bus, though as long as I have any say in the matter all your B2000 add-on cards will drop into a future 32 bit machine and work just fine. As far as comparisons to other busses. Microchannel has basically caught IBM up pretty close to the rest of the world, and the Amiga bus really was the first there (among the three).

All three systems have a method of autoconfiguration. NuBus takes a very simplistic approach; there are 16 slots, each has a ROM at the top, and each occupies 16 meg. Easy to do on a 32 bit machine, but the Mac software doesn't like this yet, so in reality each slot is 1 meg. I think Amiga's system is better in that respect, though real NuBus has a 256 meg pool to draw upon for those things that don't take up just 16M. But NuBus is better as a peripheral bus; it's slower than on-board memory, being self-clocked at 10MHz. The Amiga bus, while slower than NuBus, incurs no penalties,

speed-wise. Microchannel's configuration looks pretty convoluted, but at least it's there.

Microchannel also supports shared interrupts, something we've always had, though there's no software support for it yet, and I heard a rumor there won't be in (the initial releases of OS2) either. NuBus has one dedicated interrupt per slot, which I don't like, 'cause you can't handle system priorities right that way.

As for DMA, Amiga's DMA, except for the Coprocessor slot, is an all-or-nothing affair, whereas NuBus and Microchannel both support some fairness, so they can better deal with alternate processors on the main bus. Fairness isn't much of a win in dealing with burst DMA like hard disks, though. NuBus also supports resource and DMA locking, which is essential to multiprocessing. Microchannel and Amiga don't, though the fact that Amiga doesn't have DMA fairness kind of eliminates the need for locking, since once you have the bus, you can't be thrown off. Oh, and even on the 386 machine, Microchannel doesn't have enough bandwidth to truly support a full-time coprocessor; neither does Amiga. But then again, I haven't done MY 32 bit bus yet. NuBus could reasonably deal with one. I could rant and rave for days, but that's basically it.

(Glenn) Dave, what improvements and new features are in DOS 1.3? And, when will DOS 1.3 and the 1Mb Fat Agnes be available?

DH: Can't really get into too much of that, and it's not just because I don't really know that much about it. The next release, 1.2.1, will give you boot from arbitrary device, like hard disk or EtherNet. But no enhancements.

I think the main idea behind 1.3 will be to hone the system. 1.2 pretty much fixed the major bugs, and put in the hooks for things like RamKicked library modules. 1.3 would certainly

APL.68000

\$99

A HIGHLY OPTIMIZED ASSEMBLER BASED APL INTERPRETER FOR FAST AND POWERFUL PROGRAMS. FEATURES A COMPLETE INTERFACE TO THE AMIGA ENVIRONMENT WITH PULL-DOWN MENUS, REQUESTER AND ALERT BOXES, SPEECH, SOUND AND GRAPHICS FACILITIES.

Order direct for \$99 + \$7 shipping, \$10 Canada. VISA/MC/AMEX + 4% NJ res + 6% sales tax.

SPENCER ORGANIZATION, INC.

P.O. Box 248 Westwood, N.J. 07675
(201) 666-6011

fix all known bugs, and I think they'll be speeding things up all over. Lots of folks know about the faster print drivers. They're called 1.3, which tells me that they're targeted for the 1.3 release, which I wouldn't expect for maybe 6 months or more. But also they're very far along, so you might see them sooner. You could run just about any 1.3 enhancement that I've heard of on a 1.2 system, so there's no absolute need to release the entire thing at once, only the desire to keep OS releases well understood and under control. But as I implied, I'm not really the guy to ask for this.

(Karl Sparklin) Other than those mentioned in responding to Bill, what other changes were made to the A2000 to arrive at the B2000? I've not heard much about it.

DH: Well, the 100 pin backplane changes, which were minor, were put in for the A2000. And I didn't do anything to those, 'cause we wanted to keep the 100 pin bus as standard as possible, since we're planning to use it for a long time to come. What I did change was the coprocessor and video slots. Part of the reason for the changes was what we had identified as the reason for these slots at the time of my design, versus what was originally envisioned.

(continued)

The coprocessor slot started out being basically a local bus access point for the extra 512K of A2000 RAM, and possibly a bus-monitor type MMU at some future point. (All this does is sit around and yell, via BUS ERROR, if an access violation occurs. No translations possible.) Anyway, the extra 512K was gone when I put that on the motherboard, and there was a need for a real 68020 upgrade, which includes a real MMU. So I added the B2000 coprocessor protocol stuff, which basically allows a 68020 card in that slot to request the bus from the 68000 and become a fully equivalent bus master. Any 100 pin bus masters completely lock out all others when they have the bus, as I mentioned before. Okay for burst stuff, but not when you're taking control. Anyway, it is possible for a 68020 or whatever to come on, run as a full 68000 replacement with hard disk DMAs and everything, then get off (though our 68020 card never gives the bus back, for A2000 compatibility). On the A2000, you have to pull the 68000 out to use a coprocessor, but I kept the data sense of the bus arbitration lines the same, so a single jumper on our '020 card sets it up for the proper motherboard.

The other main enhancement I added was the second half of the video slot. Originally, that was intended for genlocks or modulators, but I thought it was kind of weak, so I added the second slot for access to all 12 bits of digital video and more system clocks. George Robbins basically suggested the port lines from the parallel port go there too, so we can have some communication to devices there. Anyway, consider the video slot a standard thing that you'll see in future machines, while the coprocessor slot is a machine specific thing that will probably go away if the general expansion bus gets upgraded in the future on a 32 bit machine. Huff and puff, my fingers hurt.

(Paul B.) Dave, we have been hearing for some time now of a new Agnes chip that will give us extra chip

memory, higher resolution, more colors, etc. Is there any truth to this at all? And if so, when?

DH: There's certainly work being done on new chips all the time. First of all, anyone who takes a second look at the Fat Agnes architecture will notice that it's controlling 1 meg of memory right now, though internally Agnes only talks to 512K of RAM as chip memory. So it's definitely possible that a drop-in replacement would support 1 meg of CHIP memory. However, when such parts will be generally available, I can't say. When I get mine, then maybe we can talk [Grin]. As far as higher resolutions and more colors go, that is more complicated, because it requires an agreement between Agnes and Denise. Agnes supplies the video sync signals and stuff, but Denise is responsible for interpreting the information that Agnes gives it and turning it into the video you get. From what I know of Denise, higher resolution is possible but also not very straightforward; more colors is more difficult. Not that you won't ever see such things, but they're not going to be here real soon or anything.

(Isaac Cruz) Dave, given that the A500 power supply is, to put it mildly, barely functional, are there any plans to provide a more robust supply either by C-A or other vendor?

DH: Progressive Peripherals is making some kind of memory expansion for the A500, and it comes with a beefier power supply. That's the only one I've heard of so far; I'd expect most add-ons to have their own add-on supplies instead, since that's simpler to do unless you go inside.

(Jim Williams) Dave, what is the supply of Amigas like? Will there be enough for Christmas sales, since dealers seem to be getting a lot of back orders?

DH: I sure hope so. Last I heard, they're just barely keeping up with A500 demand, but they are keeping

up. A2000s are coming along slower, I guess due to the fact it's a newer machine, there are more parts to gather from the four corners, and the trade-in deal has probably boosted demand more than CBM expected. And the fact that the Germans, at least, are buying them at more a consumer than a business/professional machine rate. Which suits me just fine, if only they can get production up. By the way, they are building them downstairs in West Chester, so I expect that the dealer supplies, now pretty well dried up, should start flowing again soon.

(Thom C.) I missed out on Comdex but am thinking about on going to the World of Commodore show. Do you think the trip would be worthwhile?

DH: I think so, Thom. I was at WOC last year, and, based on that, I expect that CBM will be showing pretty much the same kind of things that were at Comdex, probably with the addition of C-64s and C-128s. I'll have a 68020 board for them. The WOC show was a step above CES last year, my first trip to WOC. I left feeling all kinds of warm and fuzzy about Amigas.

RR: I'll close this out with a question from Marlene, who's doing duty on another channel. Dave, what about 68030 support?

DH: Good question. The current OS would detect a 68030 as a 68020, and run it accordingly. A reasonable thing to do, since the 68030 wasn't fully specified when 1.2 was frozen. I'm sure 1.3 will have a 68030 flag, though.

RR: Marlene tells me that what she REALLY wanted me to ask is [Grin] do you think CBM can remain competitive, especially with business oriented users, if you don't pursue the 68030 rapidly (since by the time the 68020 support is available, that chip will be "old news")?

DH: Fair question. Let's look at it from another point of view. The 68000 was already in two machines on the market, one with a significant lead time (the Mac), by the time the A1000 came out. Even if Apple has a 68030 done by the time we have a 68020 machine, I'm skeptical that it'll be that much of a difference. Then again, I still don't see the 68030 as THAT MUCH of a great advantage over the 68020 in our future designs, given the same system clock speed. Also consider the fact that a 68030 will add maybe \$800 or more to the final cost of any 32 bit machine we might produce. About all I can say is that, (1) our 32 bit machine will do things that Apple's doesn't, (2) we certainly won't ignore the 68030, now that it's real, and (3) at least WE have an OS that supports these 32 bit chips. And, on the '020 vs. '030 question, we may have a surprise or two you aren't considering.

RR: I, for one, have a dozen more questions, and I know you all do too ... but Dave's been typing (LOTS!) for about two hours, and that takes a toll. So, I'm going to close down the official CO. You are all welcome to chat for as long as you wish. Dave, thanks a million for a great CO and you have a standing invite to come back! Thanks!

[At this point, the formal conference was terminated. The following are excerpts from the informal chat which followed.]

(Paul B.) When will we see a 32 bit machine, Dave?

DH: Well, I've got this '020 board to finish up. "Not soon enough" is a good answer.

(Doug Erdely) Any word when the 68020 board for the A2000 will be out? And the price?

DH: Don't know the price or release. It's almost done, engineering-wise.

(Nelson) Will the A1000 support 1.3 and vice-versa?

DH: Yes, 1.3 will be available on the A1000.

(Marlene/SYSOP) Dave, you must be back-ordered like crazy on the 2000.

DH: I'm worried, the A2000 seems so back-ordered from what I hear. I'm still waiting for MY A2000 (at home, got one at work).

(Thom C.) BTW, Dave, have you guys had any problems with the seating of the Bridge card in the slot of "your" 2000?

DH: I've never had any bridge seating problems, other than it's generally tough to plug in. I've drawn blood more than once pulling them out.

(Marlene/SYSOP) Dave, have you heard of some 2000s not GURUing properly? Mine does this; the screen just freezes, and I have to control-A-A to get a GURU.

DH: There is a know Alert() bug. Happens on any system with \$C00000 memory.

(Glenn) Dave, has C-A recognized yet that there is some kind of problem with SCSI and the A2090?

DH: C-A has recognized that there is some kind of problem with SCSI in general: that it's a non-standard standard. They're trying to keep a list of all SCSI devices that work properly.

(Glenn) Will there be a software fix, so that the SCSI formatting problems go away?

DH: Glenn, there apparently should be a deep-format capability, which, as I understand it, is basically a utility that sends the "format this thing" command to the SCSI controller. What is absolutely needed, for this and other things, is a SCSI.DEVICE.



Oriental Desk Top Art
vol. 1



- High resolution images
- Oriental artwork
- IFF format
- Use as clipart in desktop publishing programs

Price: \$29.95

Software Integration Solutions

16496 Bernardo Center Drive
San Diego, CA. 92128

Tel: 619-451-3094




The main problem with SCSI is its implementation. The thing is that some controllers are basically dumb SCSI controllers. But SCSI is a smart interface standard. So on these devices, it's up to the implementor to provide a microcontroller with the smarts. If there's a bug in the SCSI code of the microcontroller, you lose. The A2090 uses a Western Digital SCSI chip with intelligence in it. So if we screwed up the SCSI, at least low-level stuff, that's them, not us. But I'm pretty confident that WD did it right.

I'm really happy with my A2090. I've got two ST-506 drives on the one at work, two at home. It's real DMA, reasonably priced, and even if you get ST-506 drives, I think SCSI will become increasingly important in future years.

(Steve/SYSOP) Glenn, SCSI works fine with the Adaptek 4070, but there is a DEFINITE problem with the driver if you have a non-workbench screen up. Using a SCSI drive you get I/O errors ALL the time; hit retry and it recovers, go back to the non-WB screen and more R/W errors. (This happens with 1 ST506 connected and 1 SCSI.)

(continued)

HARD DRIVES

150 meg	\$1595.95
65 meg	\$ 949.95
40 meg	\$ 879.95
20 meg	\$ 659.95

Hard drive comes complete with SCSI controller, fan cooled power supply with case and a true SCSI hard disk (no interface boards).

Full ONE YEAR warranty!

**ASK ABOUT OUR REMOVABLE
MASS STORAGE DEVICES**

EXTERNAL 3.5 DRIVE ONLY \$ 159.95

CABLES: Serial or Parallel **\$ 9.95 each**

RAM

AMIGA 500

512K with CLOCK ..	\$ 129.95
1.5 meg INTERNAL .	\$ 389.95
2 meg EXTERNAL ...	\$ 449.95

AMIGA 1000

1.5 meg INTERNAL .	\$ 389.95
2 meg EXTERNAL ...	\$ 449.95

AMIGA 2000

2 meg INTERNAL ...	\$ 409.95
--------------------	-----------

ASK FOR OUR COMPLETE PRODUCT CATALOG

PIONEER COMPUTING

-BLAZING NEW FRONTIERS IN PERSONAL COMPUTING-

2469 EAST 7000 South #200
Salt Lake City, UT 84121

(801) 942-1174

DH: What Steve's talking about is based on what I call DMA latency. You can eliminate some of this by running SlowMemLast before Binddrivers if you've got fast memory on the bus. But what Steve says makes me believe that there must be a software problem too, for at least I was led to believe that SCSI was happier with choppy DMA access than ST-506.

RR: Dave, what exactly is the source of the problem?

DH: The problem is based on the bandwidth of CHIP RAM getting tied up in display at higher resolutions. The DMA device asks for the bus, and the 68000 would like to oblige. It gives the device a Bus Grant, which says, "Things are all right by me for you to take over the bus." However, the Amiga DMA arbitration follows the 68000 protocol very closely, and this says that you must wait until the end of a memory cycle before you take

over the bus. The custom chips keep the 68000 off the bus by wait stating it, which means you may have to wait a full scan line before your DMA gets the bus, even though you really have the expansion bus yourself.

The problem is compounded when you're DMAing to Agnes controlled memory, because you not only have this long wait for control of the bus, but you then have to spit out DMA data only during non-display time. The end result is that you can easily have a disk overrun. If you DMA to expansion bus memory instead, once you have control over the bus, you get your burst DMA without waiting, since the custom chips don't wait-state anything that's using expansion bus memory. Well, the problem is that in hi-res modes, ALL of the video bus bandwidth is being used during display time. I would have liked to at least remove the latency problem, but the obvious solution to the problem requires an 010 or 020.

(Murray Trickett) Are there any boards around in the USA that allow you to plug in IBM type hard drive to the A500?

DH: There's some kind of device/driver combination I've read about that turns a PC hard drive into an Amiga hard drive, without a bridge card. Don't think it was specifically aimed at the A500, though they may have a version.

(KEITH YOUNG) Dave, is the 1.2 DOS in the A500/A(B)2000 any different than that released (on disk) for the A1000?

DH: The 1.2 KickStart and ROM are identical. The A2000/A500 vs. A1000 WorkBenches have some minor differences on them.

Copyright 1987 by CompuServe
AmigaForum, Dave Haynie.

•AC•

The AMICUS Network

New Public Domain Disks, Commodore and Developers

By John Foust

The national computer networks have been very busy during the past few months. This activity brings a wind-fall of new public domain software. So, here are descriptions of three new AMICUS disks, numbers 24 to 26, as well as highlights from the latest Fred Fish disks, 103 to 110.

Disk 24

AMICUS disk 24 has a little of everything, from C, assembler, BASIC and Modula-2 program examples, to games, animations, and utilities.

Sectorama is a disk sector editor for any AmigaDOS file-structured device. David Joiner of MicroIllusions developed it for recovering files from a trashed hard disk.

Iconize reduces the size of IFF images, performing full anti-aliasing for the best results possible. It does a wonderful job and comes with a companion program, Recolor, which remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, you can make icons which look like miniatures of the pictures.

CodeDemo is a Modula-2 program which converts assembler object files to inline CODE statements. Modula-2 programmers would use this tool to incorporate short assembly language programs in their programs. CodeDemo comes with a screen scrolling example that makes a fly walk across the screen. AmiBug is a cute little

Workbench hack that makes the same fly walk across the screen at random intervals. Otherwise, it is completely harmless.

BNTTools are three examples of assembly language code from Bryce Nesbitt. The programs include: SetLace, a program to switch interlace on and off; Why, a replacement for the AmigaDOS CLI Why command; and LoadIt, a program to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)

Monolace is a CLI program which resets Preferences to several colors of monochrome and interlace screens. C source is included so you can change the preset colors. It works with DisplayPref, a CLI program which displays the current Preferences settings.

BoingMachine is the latest animation from Ken Offer, a winner in the BADGE Killer Demo contest. Offer makes his animations with Sculpt 3D. BoingMachine is a ray-traced animation of a perpetual motion Boing-making machine. It includes the latest version of the Movie program, which has the ability to play sounds along with the animation.

Daisy is a minimal example of using the translator and narrator devices to make the Amiga talk. It is written in C. QuickFlix is a script-driven animation and slideshow program that flips through IFF images. BMon is a system monitor program written in AmigaBasic; it lets you perform simple manipulations of memory.

Moose is a background program like the Macintosh Talking Moose. At random intervals, a small window opens with a moose that resembles Bullwinkle. The moose says something witty, such as "Get back to work," and the window disappears. You can change the phrases, if you like.

DGCS, short for "Deluxe Grocery Construction Set," is a simple Intuition-based program for assembling and printing a grocery list.

The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe. This insidious program was detailed in *Amazing Computing* V2.12. The program has been fully disassembled by Commodore technical support, and Bill Koester's full explanation of the virus code is included. According to Koester, "Just like the real thing, the best course of action is education and prevention." One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

The Commodore analysis of the virus code says it only prints a message at random intervals. Although the current version of the virus does not format or destroy most disks, it can harm some copy-protected software, if the software has special program code stored in the boot block of the disk. It is interesting that this virus can only harm programs copy-protected in this way.

(continued)

Disk 25

Nemesis was an entry to the BADGE Killer Demo contest. It is a graphics demo that pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.

The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADD-MEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.

KeyBird is a BASIC program that edits keymaps. You can adjust the Workbench keymaps or create your own.

8ColorWB modifies the Workbench so three bitplanes are used, meaning icons can have eight colors, instead of four. Several eight-color icons are included. Fortunately, the public domain program "zapicon" or "brush2icon" gladly converts eight-color IFF brushes to icons, so you can use Deluxe Paint to make icons for this new Workbench.

BrushIcon converts brushes to icons ... or maybe it isn't. The documentation is in a strange foreign language with lots of umlauts and long noun-based words, such as "Workbenchomgvinngen."

Egraph is a graphing program which reads [x,y] values from a file and displays them on the screen. It is similar to the same-named Unix program.

Keep 1.1 is a message-managing program for telecommunications fans. It lets you save messages from an online transcript to another file. Keep 1.1 understands the message format of the national networks and several types of bulletin board software. It moves through the transcript and lets you save messages that interest you.

Kill.fastdir is a must-have for CLImate owners, written by CLImate's authors. To speed up directory access, it creates a small file in each directory on a disk which contains the information about the files. Some CLImate owners don't like all the "fastdir" files, though, so this program removes them all, from each directory.

The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.

PW_Utility is a shareware utility for ProWrite users that changes margin settings and font types. Guru is a CLI program that prints out probable causes for Guru mediations; C source is included. DiskWipe, the latest from the Software Distillery, removes files from directories or disk drives, much faster than "delete." Snow is an AmigaBasic program that makes snowflake designs. Mlist is a mailing list database. This disk also has a softball statistics program to maintain records for a team.

Dodge is a short Modula-2 program which moves the Workbench screen around after a period of time, presumably to prevent the edges of your monitor screen from burning out.

Disk 26

At press time, a package arrived with Todor Fay's SoundScape module code from his *Amazing Computing* articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules. AMICUS 26 also includes Fay's ImageMaker program, an interesting tool which edits Image structures for C. It loads and saves C code directly.

Claz2 is an update of the program which converts IFF images to PostScript files for printing on laser

printers that understand particular page description language. SDBackup is a hard disk backup program with Lempel-Ziv compression to reduce the necessary number of disks. TCB prints information about tasks and processes in the system; assembler source is included.

FunBut lets a function key act like a rapid series of left mouse button events. The author says this program is great for winning Defender of the Crown.

DC is a handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. DC is a Workbench program that sends a DiskChange signal to the operating system. Instead of typing "diskchange df2:" over and over again in the CLI, just click on the icon. C source is included.

This disk also has a system configuration file that makes the screen wide enough to show 80 columns of text in the Scribble! word processor. Two programs are also included to move the Scribble! spelling dictionary to and from the RAM disk.

Lexical analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability. HexDump is a Modula-2 program to display memory locations in hexadecimal. Tartan is an AmigaBasic program to design Tartan plaids. DirMaster is a disk catalog program.

BMP plays 8SVX sampled sounds in the background while something else is happening in the Amiga. The author suggests playing sampled sounds as your Amiga is booting, for example.

This disk also has a collection of mouse pointers, along with a Workbench program to display them in turn. A CLI program changes your pointer to a given pointer.

New Fish

Fred Fish has added disks 103 to 110 to his collection. I must report that Fred Fish is a real person. I've talked to many Amiga owners who thought the name of this disk collection was an imaginative invention.

Disk 103 contains: a C example of AVL trees, an efficient data structure technique for some applications; an RPN calculator; a C cross reference generator; DosKwik, a program that quickly saves and restores a RAM disk; IntuiDOS, an alternative Intuition interface to the CLI; and a text import utility for Microfiche Filer.

Disk 104 contains a spreadsheet program called AnalytiCalc. It seems very powerful. AnalytiCalc supports large spreadsheets and has many functions found in most spreadsheet programs, as well as some unique features that make it more like an outline processor or a database program. C source is included, and the author plans other extensions to the program.

Disk 105 has a number of assembly language programs for manipulating memory and CLI tools, some AmigaBasic programs for solving least squares problems, and a Freudian analysis program. Bison is an update to compiler generator tool on disk 51. FlamKey password-protects your Amiga. If the correct password is not entered, the keyboard and mouse are locked out. Pere-et-Fils is an example of reentrant processes. Record and Replay are updates to the mouse and keyboard recording programs on disk 95.

Disk 106 has an update to the function key programmer from disk 89 and some IFF pictures. QuickFlix, a slideshow and animation program, and a game called RistiNolla are included. RistiNolla resembles gomoku, if you know what that is.

Disk 107 contains several FORTRAN programs. Two programs plot data on the screen, in two and three dimen-

sions. Another program drills young people in multiplication skills. A program is also included to slowly copy a text file to the printer, in order to avoid overrunning its buffer. Another program provides an example of timing events.

107 also includes a program to convert text files of numbers to Lattice UniCalc spreadsheet format, and tutorials on backing up hard disks using public domain utilities and ways to avoid interlace flicker.

Disk 108 has MonIDCMP, a program to monitor IntuiMessages that pass through an IDCMP window. (If you haven't guessed, this is a programmer tool.) A program to calculate the characteristics of steam (useful for engineers) and Tek, an update to the Tektronix graphics terminal emulator on disk 52, are also included.

Disk 109 has a new animation called "Machine" from VideoScape author Allen Hastings. Machine is a scene from a factory that makes Boing balls. This disk also has a CP/M simulator which, like the Transformer, interprets the machine language of a CP/M computer and pretends to run CP/M programs as if they were on a H-19 terminal.

UUpc is a series of programs that lets your Amiga function as a Usenet node. You can send and receive Usenet mail and news, including stuff from the ever-popular "comp.sys.amiga" area. This software may tempt people who would like Usenet to come to them, instead of them going to Usenet. Be forewarned, though: This program requires a bit of hackerish tinkering and cooperation with the administrators of a local Unix site. C source is included.

Disk 110 has an update of the C compiler on disk 53 which generates assembler source code in Metacomco form. A public domain assembler, written in C, is provided on this disk, along with "blink," a freely distrib-

utable linker. You need the "amiga.lib" from the Commodore developer kit in order to write complete Amiga programs, however.

FAUG meeting

In November, I attended my first FAUG meeting. FAUG stands for "First Amiga User Group." It is headquartered in the Silicon Valley area where the Amiga was born, and meetings are now held in the Hyatt Hotel in Palo Alto, California. On December 1, FAUG will hold its second anniversary meeting. A cash bar is available in case the speakers are boring.

(I retain the belief that the AMICUS Network was formed before the First Amiga User Group, but I've never challenged its name. Of course, I might not win the argument, considering that the AMICUS Network has never held a meeting, and no one, not even me, will admit to being a group officer.)

Average FAUG attendance is about 500. On the way to the hotel, I spotted several Amiga bumperstickers at the intersection in front of the hotel. Can you imagine traffic and parking problems because of a user group meeting?

The speakers at FAUG are rarely boring. In fact, FAUG group leaders complain that they are having a hard time impressing the group, because most Amiga notables have already spoken at the meetings. They don't know what to do for the anniversary meeting because the users have seen it all.

There were many familiar faces in the crowd. Dan Silva, author of Deluxe Paint, many of the original Los Gatos Amiga people—from Jay Miner on down—and many current Amiga developers were present. I spotted at least a dozen developers, which is more than I've seen at some trade

(continued)

shows. Allen Hastings brought along videotapes of his latest VideoScape animations.

Politics and Problems

At COMDEX, I thought Commodore was playing a silly definition game. In later discussions, one topic reigned supreme: the high resolution monitor and the 68020 card. The question was, "Did Commodore announce products, or was it just showing them?" The answer measures Commodore's adherence to a policy decision. This decision promised to prevent the unpleasant reaction when products are announced too far in advance of actual shipment.

When Commodore showed the Sidecar, A500, and A2000 well in advance, some dealers complained. Many people heard about the new hardware or read "product previews" in magazines and came to dealers expecting to purchase the new computers. When the dealers told customers the computers were not shipping, some felt misled. Dealers do not like unhappy customers, particularly if the customers are unhappy about something Commodore did. Word about unhappy customers trickled back to Commodore, and they came up with a solution and revealed it at the next show: They would not announce products before they were ready to ship.

The silly game comes down to the difference between "announcing" and "showing" a product. When Commodore shows a device, but says little about it, is it an announcement? Is it only an announcement when a ship date is included?

Without any official statement regarding a new device, no one can spread more than a rumor. If you can say you "saw it at COMDEX, in the Commodore booth," stories gain credibility. If you can only say "I heard it somewhere," stories are taken as just that—only stories.

Very little information was supplied at COMDEX regarding the high resolution monitor and the 68020 card. There was no mention of these things in the press releases, and there were no flyers.

With no official information, misinformation reigns. Before I checked my sources, I had some serious misconceptions about the high resolution system. A source who should have been reliable told me the card was for the Amiga 2000. In truth, it is a special monitor coupled with software running on any Amiga system.

I tend to think little was accomplished by the lack of information about the new products. The word will get back to the same people who waited for Live!, the Sidecar, A500, and A2000. They may complain just as much—unless the products appear soon.

Booth Space

At COMDEX, Infinity Software was asked not to show Shakespeare (their desktop publishing program) because Gold Disk was showing Professional Page. CSA was not allowed to exhibit in the Commodore booth because their 68020 products would draw attention away from Commodore's own 68020 board. I had visions of CSA representatives being turned away on the show floor, arms full of product boxes.

Booth space is premium at any show. Commodore must decide who can display in their booth. In a 75 foot square booth, there is no room for duplication. It is a very cut-and-dried process—one or two desktop programs, one spreadsheet, one CAD program, one or two video products, and a few games.

The situation is scary, given the previously cordial relations between Commodore and CSA. Commodore always welcomed CSA into their booths. The action against CSA may frighten other vendors.

When dealing with any outsider, Commodore must be necessarily tight-lipped about research. Unfortunately, they are just as tight-lipped with information for third-party developers. The situation is potentially tense when the Commodore product in development is similar to a product from a third-party developer. The third-party developers' perspective is easy to imagine. Why is Commodore nosing in on territory the developers consider theirs? Why should the parent computer company compete with third-party developers?

Sculley on Developers

I have wanted to use the following quote for several months. It seems particularly appropriate in light of the incidents at COMDEX. The quote is from John Sculley, CEO of Apple Computer, in the September 1987 issue of *Playboy* magazine. Sculley has been interviewed recently in dozens of publications to promote his latest book. I don't agree with all of his ideas, but this quote rings true, especially when considered as advice to Commodore. At this point in the interview, Sculley is discussing the informal network of business people that surrounds any large company.

"What I've discovered—and I hadn't known this until I came to Apple—is that third-wave companies are totally dependent upon the network of third-party companies they can build around them. This allows us to think of our personal computers as platforms on which others can go and innovate; and in fact, our experience is that the best innovation has not been done by us but by outsiders.

"We didn't invent VisiCalc, as you know. IBM didn't invent Lotus 1-2-3. We didn't invent Pagemaker. Those were all done by entrepreneurs... The impact on almost any market that Apple has been in has been caused by a network of resellers or a network of enthusiasts or a network of developers who are doing things with the platform technologies that Apple creates."

Texts: FrntKeys explains how to read function keys from Amiga Basic. HackerSh explains how to win the game "hacker" by installing a 68010 in your Amiga sending escape sequences to your printer tips on setting up your startup-sequence file list of Transformer programs that work Printer Drivers: Printer drivers for the Canon PJ-1080A, the C Job Prowler, an improved Epson driver that eliminates streaking, the Epson LO-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P100x family, and the Smith-Corona D303, with a document describing the installation process. AMIGUS Disk 12 Instrument sound demo This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a bongo, a callopie, a car horn, claves, water drip, electric guitar, a flute, a sharp spigot, a koddum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a star, a snare drum, a steel drum, bells, a vibraphone, a violin, a wailing guitar, a horse whinny, and a whistle. AMIGUS Disk 11 C programs drutil intuition-based, CLI replacement manager S-E shows and adjusts priority of CLI processes, S-E ps shows info on CLI processes, S-E vidcat displays Composite RLE pics, S-E AmigaBasic programs pointedit pointer and sprite editor program optimize optimization example from AC article calendar large, animated calendar, diary and date book program amortize loan amortizations brushesBOB converts small IFF brushes to AmigaBasic BOB OBJECTS grids draw and play waveforms hilbert draws Hilbert curves radio making radio story generator mailtalk talking mailing list program meadows3D 3D graphics program, from a CPM article mousetrack mouse tracking example in hires mode slot slot machine game slotact the game switch pachinko-like game weend makes strange sounds Executable programs cp unix-like copy command, E cl screen clear, S-E clis unix-like screen editor uses 'tiff' output to fix files pm chart recorder performances indicator Assembler programs cls screen clear and CLI arguments example Module-2 moving worm graphics demo trails converts Module-2 keywords to uppercase caseconvert Bresenham circle algorithm example Furn 12 templates for the spreadsheet. Analyze Analyze There are four programs here that read Commodore 64 pictures files. They can translate Koda Pad, Doodie, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part. AMIGUS Disk 12 Executable programs blink 'blink' compatible linker, but faster, E-D clean spins the disk for disk cleaners, E-D epsonset sends Epson settings to PAR from menu E-D showimg view hires pics in low-res superbmap, E-D speakeime tell the time, E-D undecore undecore a file, E-D crnaphidm converts Apple II low, medium and high res pictures to IFF, E-D menued menu editor produces C code for menus, E-D quick quick disk-to-disk nibble copier, E-D quickEA copies Electronic Arts disks, removes protection, E-D bed 1.3 demo of text editor from Microsmiths, E-D C programs spn3 rotating blocks graphics demo, S-E-D popdi start a new CLI at the press of a button, like Sockk, S-E-D vsprite VSprite example code from Commodore, S-E-D AmigaBBS Amiga Basic bulletin board prog., S-D Assembler programs sta:10 makes star fields like Star Trek intro, S-E-D Pictures MountMandelbrot 3D view of Mandelbrot set Star Destroyer hires Star Wars starship Robot robot arm grabbing a cylinder Tests vendors Amiga vendors, names, addresses carcd fixes to early Carcd memory boards cinclud cross-reference to C include files mindwalker clues to playing the game well slideshow make your own slideshows from the Kaleidoscope disk AMIGUS Disk 13 Amiga Basic programs Routines from Carolyn Scheppler of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFF program. With example pictures, and the SawElBM screen capture program. Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Faust for Applied Visions. With	documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound. Executable programs gravity Sci Amer Jan 86 gravitation graphic simulation, S-E-D Tests MIDI make your own MIDI instrument interface, with documentation and a hi-res schematic picture. AMIGUS Disk 14 Several programs from Amiga Computing issues: Tools Dan Kay's C structure index program, S-E-D Amiga Basic programs: BMAP Reader by Tim Jones FFBush2BOB AutoRequester DOSHelper Windows help system for CLI controls, S-E-D PETrans translates PET ASCII files to ASCII files, S-E-D C Squared Graphics program from Scientific American, Sept 86, S-E-D drif adds or removes carriage returns from files, S-E-D dpcode decodes Deluxe Paint, remap ves copy asks Yes or No from the user, returns exit code, S-E queryWB VioCalc type spreadsheet, no mouse control, E-D vc views text files with window and slider view Org, Spring, ysBong, Zing are sprite-based slider Bong style demos, S-E-D QClock sClock, wClock are window border clocks, S-E-D Tests An article on long-persistence phosphor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga. AMIGUS Disk 13 The C programs include: pr a file printing utility, which can print files in the background, and with line numbers and control character filtering. tm displays a chart of the blocks allocated on a disk 'ask' questions an 'execuete' file, returns an error code to control the execution in that batch file 'Sta' an enhanced version of AmigaDOS 'status' command. Dissolve' random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion. PopCLIZ' invoke new CLI window at the press of a key. The executable programs include: Form' file formatting program through the printer driver to select, print styles DiskCat' catalogs disks, maintains, sorts, merges lists of disk files 'PSound' SunRize Industries' sampled sound editor & recorder 'Iconmaker' makes icons for most programs 'Fractal' draws great fractal seascapes and mountain scenes '3D Breakout' 3D glasses, create breakout in a new dimension 'AmigaMonitor' displays lists of open files, tasks, devices and ports in use. 'Comoids' version of 'laterside' for the Amiga. 'Sizzers' high resolution graphics demo written in Module 2. Tests: 'task:bt' explains escape sequences the CON: device responds to. 'FKey' includes template for making paper to set in the tray at the top of the Amiga keyboard. 'Spaw' programmer's document from Commodore Amiga, describe ways to use the Amiga's multitasking capabilities in your own programs. AmigaBasic programs: 'Grds' draw sound waveforms, and hear them played. 'Light' a version of the Tron light-cycle video game. 'MigsSol' a game of solitaire. 'Stars' program to calculate betting averages 'Money' "try to grab all the bags of money that you can." AMIGUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah. AMIGUS Disk 16 Juggler' demo by Eric Graham, a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are fipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product. IFF pictures perodies of the covers of Amiga World and Amazing Computing magazines. C programs: 'InputHandler' example of making an input handler. 'FileZap3' binary file editing program 'ShowPrint' displays IFF picture, and prints it. 'Gen' program indexes and retrieves C structures and variables declared in the Amiga include file system. Executable Programs: 'Frishunk2' repers an executable program file for expended memory 'ms2mus' converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases. 'Missile' Amiga version of the 'Missile Command' video game.	This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park. AMIGUS Disk 17 Telecommunications disk which contains six terminal programs. 'Comm' V1.33 term prog, with Xmodem, WModem, term prog, includes Super Kermit 'ATerm' V7.2 Dave Wecker's VT-100 emulator with Xmodem, Kermit, and scripting 'VT-100' V2.6 VAD(360) port of the Unix C-Kermit 'Amiga Kermit' Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression 'AmigaHost' V0.9 for CompuServe. Includes RLE graphics abilities & CIS-B file transfer protocol, expansion memory necessary 'FixLink' removes garbage characters from modern received files 'FixQ' filters text files from other systems to be read by the Amiga E.C. 'Tat' executable version for use with mem expansion article in AC V2.1 'addmem' file documentation and a basic tutorial on un'loading files 'arc' for making 'arc' files E.C. AMIGUS Disk 18 Logo Amiga version of the popular computer language, with example programs, E-D TVText Demo version of the TVText character generator PageSetter Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package. FullWindow Resizes any CLI window using only CLI commands, E-D Life3d 3-D version of Conway's LIFE program, E-D Defdisk CLI utility to re-assign a new Workbench disk, S-E-D Calendar WKS Lotus-compatible worksheet that makes calendars SetKey Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D VPG Video pattern generator for aligning monitors, E-D HP-10C Hewlett-Packard-like calculator, E-D SetPrefs Change the Preferences settings on the fly, in C, S-E-D StarProbe Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D ROT C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D Scat Like Igg, windows on screen run away from the mouse, E-D DK Decays' the CLI window into dust, in Module 2, S-E-D DropShadow2 Adds layered shadows to Workbench windows, E-D AMIGUS Disk 19 This disk carries several programs from Amiga Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hires image of Andy Griffith, and five Amiga Live pictures from the Amiga Stories episode that featured the Amiga. Solve Linear equation solver in assembly language, S-E-D Gadgets Bryan Catey's AmigaBasicBarnal, Bryan Catey's AmigaBasic household inventory program, S-D Household Jim Shanks' Waveform WorkbenchBasic, S-D Waveform John Kanner's AmigaBasic disk librarian program, S-D DiskLib Ken Smith's AmigaBasic subscript example, S-D Subscripts C programs and executables for Hamlet Maybeck Tolly's intuition tutorials, S-E-D String, Boolean Bob Reinerman's example for making small C programs, S-E-D Skinny C Make C look like COMAL header file, Makes Emacs function key definitions by Greg Douglas, S-D COMALh Snoop on system resource use, E-D EmacsKey Ben's Tale character editor, E-D Almon 1.1 CLI program shows the size of a given set of files, E-D BTE CLI window utility resizes current window, S-E-D Size WinSize WinSize CLI window utility resizes current window, S-E-D AMIGUS Disk 20 Compiler, Decoder Steve Michel AmigaBasic tools, S-D BobE BOB and sprite editor written in C, S-E-D SpriteMaster Sprite editor and animator by Brad Kiefer, E-D BitLab Bitler chip exploration C program by Tomas Rokicki, S-E-D FFC Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D Bankin Complete home banking program, balance your checkbook! E-D cons Console device demo program with supporting macro routines. freemap Creates a visual diagram of free memory inputdev sample input handler, traps key or mouse events	joystick Shows how to set up the game port device as a joystick. keyboard demonstrates direct communications with the keyboard. layers Shows use of the layers library mandelbrot FF Mandelbrot to right joystick port mouse hooks up mouse to right joystick port one.window console window demo parallel Demonstrates access to the parallel port, opening and using the printer, does a screen dump, not working printer Printer support routines, not working print.support sample process creation code, not working protect demos split drawing regions region sample font with info on creating your own samplefont Demos the serial port serial Creates 320 x 200 playfield singlePlayfield latest version of cute speech demo speechdemo simplified version of speechdemo, with IO requests textdemo displays available fonts timer demos timer device use trackdisk demos trackdisk driver AMIGUS Disk 21 Target Makes each mouse click sound like a gunshot, S-E-D Sand Simple game of sand that follows the mouse pointer, E-D PropGadget Harriet Maybeck Tolly's proportional gadget example, S-E EHB Checks to see if you have extra-half-bright graphics, S-E-D Piano Simple piano sound program CeScripts Makes cell animation scripts for Aegis Animator, in AmigaBasic This disk has electronic catalogs for AMIGUS disks 1 to 20 and Fun disks 1 to 80. They are viewed with the DiskCat program, included here. AMIGUS Disk 22 Cycles Light cycle game, E-D Show_Prtill Views and prints IFF pictures, including larger than screen PrnDrvGen2.3 Latest version of a printer driver generator Animations VideoScope animations of planes and bong ball Garden Makes fractal gardenscapes BaseSorts Examples of binary search and insertion sort in AmigaBasic AMIGUS Disk 23 An AMIGUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga Instruments a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba List INSTR program to list the instruments DMCS will not load as well as list the origins for any instrument Music a collection of 14 Classical pieces 1812Overture The 16 minute classical feature complete with Cannon! Three Amiga Music Players: SMUSPlay MusioArt2SMUS MusioStudio2SMUS
Fred Fish Public Domain Software			
Fred Fish Disk 1: amigademo Graphical benchmark for comparing amigas. amigaterm simple communications program with Xmodem balls simulation of the "kinetic thingy" with balls on strings colorful Shows off use of hold-and-modify mode. dynstone Dynstone benchmark program. dotty Source to the "dotty window" demo on the Workbench disk. freedraw A small "paint" type program with lines, boxes, etc. gad John Draper's Gadget tutorial program ghmem Graphical memory usage display prog. halfbrite demonstrates "Extra Half-Brite" mode, if you have it hello simple window demo lattp accessing the Motorola Fast Floating Point library from C palette Sample prog. to design color palettes. trackdisk Demonstrates use of the trackdisk driver. requesters John Draper's requester tutorial and example program. speech Sample speech demo program. speechdemo Stripped down "speechdemo". speechdemo Another speech demo program. Fred Fish Disk 2: libo Object module librarian. oc Unix-like front-end for Lattice C compiler. debug Macro based C debugging package. make Machine independent. make2 Subset of Unix make command. microemacs Another make subset command. portar Small version of emacs editor, with macros, no extensions xfi Portable file archiver. gotic DECUS C cross reference utility. roff Gothic font banner printer. it A "roff" type text formatter. it A very fast text formatter			

cdbrh	A highly portable forth implementation. Lots of goodies.	fractal	fscape	gomoku	dart	MacView	Views MacPaint pictures in Amiga low or
xisp	Xisp 1.4, not working correctly.	halu	halu9000	halley	hauntedM	high	res, no sample pictures, by Scott
Fred Fish Disk 4:	Prints horizontal banner	hidden	join	laz	mandel	EvenMore	Puzzle
banner	A Boyer-Moore grep-like utility	menu	minipaint	mouse	Ortello	ties	Simulation of puzzle with moving square
bgrep	CNU Unix replacement 'yacc', not working.	patch	pena	pinwheel	gbox	ShowHAM	View HAM pictures from CLI.
bison	Another Boyer-Moore grep-like utility	random-circles	rgb	rgbtst	Roid	Saltaire	ABAS games of Canfield and
bm	DECUS grep	satellite	saletalk	shades	shapes	Spin3	Graphics demo of spinning cubes, double-buffered example.
grep	simple portable Kermit with no connect mode.	sketchpad	spaceart	sphere	suprshr	Sword	Sword of Fallen Angel text adventure game written in Amiga Basic.
kermit	Replacement CLI for the Amiga. V. 1.0	speechy	spell	superpad		Trails	Leaves a trail behind mouse, in Modula-2
MyCLI	A Mandelbrot set program, by Robert French and RJ Mical	spiral	stiper	terminal		3stars	3d version of the 'stars' program below.
mandel		termtest	tom	topography		Bgmip	Low-level graphics example scrolls bitmap with ScrollPort.
Fred Fish Disk 5:	Console device demo program with supporting macro routines.	wheels	xenos	xmstipr		DskMappers	Double-buffered animation example for BOBs and VSPs.
cons	Creates a visual diagram of free memory sample input handler, traps key or mouse events	Fred Fish Disk 14:	amiga3d	update of #12, includes C source to a full hidden surface removal and 3D graphics		MemView	Displays sector allocation of floppy disks. View memory in real time, move with joystick.
freemip		beep	beep	extracts text from within C source files		Ong	Bouncing balls demo.
inp.dev		dimensions	dimensions	demonstrates N dimensional graphics		Spring	Ong, with sound effects.
joystick	Shows how to set up the gameport device as a joystick.	liezap	liezap	update of disk 10, a file patch utility		ScreenDump	Dumps highest screen or window to the printer.
keyboard	demonstrates direct communications with the keyboard.	gtxmem	gtxmem	update of disk 1, graphic memory usage indicator		Seb	Simple database program from a DECUS tape.
layers	Shows use of the layers library	converts	converts	IFF brush files to Image struct, in C		Stars	Star field demo, like Star Trek.
mandelbrot	IFF Mandelbrot program	pdterm	pdterm	simple ANSI VT100 terminal emulator, in 80 x 25 screen		TermPlus	Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.
mouse	hooks up mouse to right joystick port	shell	shell	simple Unix 'csh' style shell		Vt100	Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function
one.window	Demonstrates access to the parallel port, opening and using the printer, does a screen dump, not working	termcap	termcap	mostly Unix compatible 'termcap' implementation.		Fred Fish Disk 24:	Support files for Gimpel's 'tint' syntax checker.
parallel	Printer support routines, not working.					Aint	PO 'aint' compatible linker, faster, better.
printer	sample process creation code, not working					Blink	Updated to FF 18 'browser', in
region	demos split drawing regions	Fred Fish Disk 15:	Bobbs	graphics demo, like Unix 'worms'		Browser	Manx, with scroll bars, bug fixes.
samplefont	sample font with info on creating your own	Clock	Clock	An eight-fold symmetry dazzer program.		Btree	b-tree data structure examples
serial	Demos the serial port	Dazzer	Dazzer	Really pretty!		Btree2	Another version of 'btree'
singlePlayfield	Creates 320 x 200 playfield	Fish	Fish	double buffered sequence cycle animation of a fish		Calendar	Appointment calendar with alarm.
speechdemo	latest version of cute speech demo	Monopoly	Monopoly	A really nice monopoly game written in Amiga C.		Less	File viewer, searching, position by percent line number.
textdemo	displays available fonts	OkdataDump	OkdataDump	Okdata MLI2 driver and WorkBench screen dump program.		NewFonts	Set of 25 new Amiga fonts from Bill Fischer.
time	demos timer device use	PolyDraw	PolyDraw	A drawing program written in Amiga C.		Pr	Background print utility, style options, wildcards.
trackdisk	demos trackdisk driver	Polyfractal	Polyfractal	A fractal program written in Amiga C.		Requester	Deluxe Paint-type file requester, with sample.
Fred Fish Disk 16:	like Unix compress, a file squasher	Fred Fish Disk 16:	FF disk	A complete copy of the latest developer FF disk		Fred Fish Disk 25:	Example of making asynchronous IO calls to a DOS handler, written by C-A
compress		Fred Fish Disk 17:	The NewTex Digi-View video digitizer HAM demo disk			ConsoleWindow	Example of getting the Inuition pointer's CON or RAW window, for 1.2, by C-A.
dadoc	analog clock impersonator	Fred Fish Disk 18:	AmigaDisplay	dumb terminal program with bell, selectable fonts		DrUtil	Walk the directory tree, do CLI operations from menus.
microemacs	upgraded version of microemacs from disk 2	Ash	Ash	Perseus C Shell-like shell program, history, loops, etc.		DrUtil2	Another variant of DrUtil.
mult	removes multiple occurring lines in files	Browser	Browser	wanders a file tree, displays files, all with the mouse		FileRequester	Lattice C file requester module, with demo driver, from Charlie Heath.
scapes	demos using sound and audio functions	MC68010	MC68010	docs on upgrading your Amiga to use a MC68010		MacView	Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Ewemden.
separal	Allows changing parallel port parameters	Multidm	Multidm	rotate an N dimensional cube with a joystick		Plot	Simple IFF reader program
serial	Allows changing serial port parameters.	PigLatin	PigLatin	SAY command that talks in Pig Latin		PopCLI	Sidex-style program invokes a new CLI, with automatic screen blanking.
setpic	quickstart based sort program, in C	Screen	Screen	Screen image printer		QuickCopy	Devport disk copiers duplicate copy-protected disks.
strip	Strips comments and extra whitespace from C source	Xisp1.6	Xisp1.6	source, docs, and executable for a Lisp interpreter.		ScrollPI	Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.
Fred Fish Disk 7:	This disk contains the executables of the game Hack V 1.0.1.	Fred Fish Disk 19:	Blackjack	text-oriented blackjack game		SendPacket	General purpose subroutine to send AmigaDOS packets.
Fred Fish Disk 8:	This disk contains the C source to Hack on disk 7.	Blackjack	Blackjack	Sides by Jay Miner, Amiga graphics chip designer, showing forwardart of the Amiga internals, in 640 x 400.		SpriteMaker	Sprite editor, can save work as C data structure. Shareware by Ray Larson.
Fred Fish Disk 9:	Draws moire patterns in black and white	JayMinerSides	JayMinerSides	test program to test the key mapping routines		Tracker	Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.
more	Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.	Keymap_Test	Keymap_Test	Find unclosed file locks, for programs that don't clean up.		TrClope 3-D	space invasion game, formerly commercial, now public domain. From Geodesic Publications.
MVP-FORTH		LookMon	LookMon	converts Amiga object code to Atari format program to recover files from a trashed AmigaDOS disk		Taze	Print total size of all files in subdirectories.
prof	a more powerful text formatting program	Fred Fish Disk 20:	AmigaTaktan	converts Amiga object code to Atari format program to recover files from a trashed AmigaDOS disk		Unifief	C preprocessor to remove given #ifdef sections of a file, leaving the rest alone. By Dave Yost
setpic	Program to toggle interface mode on and off.	AmigaTaktan	AmigaTaktan	example of the AmigaDOS disk hashing function		Vbest	VT-100 terminal test program. Requires a Unix system.
skewo	a rubik's cube type demo	Hash	Hash	Hex dump utility ala Computer Language magazine, April 86		Fred Fish Disk 26:	Unix-like 'cp' copy program
sparks	moving snake Graphics demo	Hd	Hd	Hex dump utility ala Computer Language magazine, April 86		Aop	Updated version of disk on disk 15.
Fred Fish Disk 10:	An interstellar adventure simulation game	MandelBrats	MandelBrats	Mandelbrot contest winners		Can	Manx 'csh'-like CLI, history, variables, etc.
conquest		Multitasking	Multitasking	Tutorial and examples for Exec level multitasking		DetAd	Diet planning aid organizes recipes, calories
dehex	convert a hex file to binary	Pack	Pack	strips whitespace from C source		Echo	Improved 'echo' command with color, cursor addressing
filezap	Patch program for any type of file.	PortHandler	PortHandler	simple Port-Handler program that performs. Shows BCPL environment dues.		FixHunk	Fix programs to let them run in external memory.
filezap	Strip garbage off Xmodem transferred files.	Random	Random	Random number generator in assembly, for C or assembler.		Fm	Maps the sectors a file uses on the disk.
iff	Routines to read and write iff format files.	SeMouse2	SeMouse2	sets the mouse port to right or left terminal emulator with speech capabilities, XModem		KickBench	Docs, program to make a single disk that works like a Kickstart and Workbench.
id	Minimal UNIX is, with Unix-style wildcarding, in C	SpeechTerm	SpeechTerm	Demo editor from Microsmith's Charlie Heath		Lex	Compiles Fag, Fiesch, and Knocad readability of text files.
sq.usq	file squeeze and unsqueeze	TxEd	TxEd	Full-featured drawing program by Stephen Vermeulen.		TunnelVision	David Addison ABASIC 3D maze perspective game.
trek73	Star Trek game	Fred Fish Disk 21:	McExample	This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!		Vc	Visical-like spreadsheet calculator program.
yacht	Dice game.	Fred Fish Disk 22:	Lemias	This disk contains two new 'strains' of microemacs. version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.		Vt100	Version 2.2 of Dave Wecker's telecom program
Fred Fish Disk 11:	slide show program for displaying IFF images with miscellaneous pictures	Fred Fish Disk 23:	Pemias	version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more. By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.			
Fred Fish Disk 12:	Shows a rotating 3 dimensional solid 'Amiga sign'.	Fred Fish Disk 24:	Conques	interstellar adventure simulation game update to shell on Disk 14, with built in comm and named variables substitution.			
ArgoTerm	a terminal emulator program, written in assembler	Fred Fish Disk 25:	Csh	update to shell on Disk 14, with built in comm and named variables substitution.			
arrow3d	Shows a rotating 3 dimensional wire frame arrow.	Fred Fish Disk 26:	Conques	interstellar adventure simulation game update to shell on Disk 14, with built in comm and named variables substitution.			
ld4	directory listing program	Fred Fish Disk 27:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
loraExec	two progs for launching progs from Workbench, presently only works under CLI.	Fred Fish Disk 28:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
SetWindow	Makes an icon show a second image when clicked once	Fred Fish Disk 29:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
SetAlternate	terminal emulator, with ASCII Xmodem, dialer, more.	Fred Fish Disk 30:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
Fred Fish Disk 13:	A Bundle of Basic programs, including:	Fred Fish Disk 31:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
Jpad	toybox	Fred Fish Disk 32:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
mandelbrot	xmodem	Fred Fish Disk 33:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
addbook	algebra	Fred Fish Disk 34:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
amgseq1	amiga-copy	Fred Fish Disk 35:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
bounce	box	Fred Fish Disk 36:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
canves	carol	Fred Fish Disk 37:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
colorcircles	copy	Fred Fish Disk 38:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
cuppaste	date	Fred Fish Disk 39:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
dragon	draw	Fred Fish Disk 40:	ABASIC	Amiga Basic demos from Carolyn Schepner.			
Eliza	ezerm	Fred Fish Disk 41:	ABASIC	Amiga Basic demos from Carolyn Schepner.			

Yabing	Original style game program shows sprite collision detects	ProffMacros	runtime library. Author: Matt Dillon Subset Berkeley's 'ms' and 'mm' macros for 'proff'	Law	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by William Rucklidge	MicroEmacs	Conroy MicroEmacs V3.60, newer than disk 22. S-E-D
Fred Fish Disk 37	This disk is a port of Timothy Budd's Little Smalltalk system, developed by Bill Kinnear at Washington State University.	ValSpeak	Transforms a file from English to Valley Speak.	MIDItools	Programs to play/record through the MIDI I/F. by Fred Cassiner	PeerFont	Like Topaz, but rounded edges.
Fred Fish Disk 38		Fred Fish Disk 47	Simulation of a robotic arm, very good graphics, teaching tool, including C source.	MoreFlows	Program to make the Work Bench Screen larger than normal. by Neil Kahn and Jim Mackraz	Vsories	Generates fractal scenery. S-E-D
CSquared	Sep 85 Sci American, Circle Squared algorithm	3D-Arm		Tilt	Program to make your Amiga look like it didn't pass vibration testing. by Leo 'Bois Ewhac' Schwab	Fred Fish Disk 52	This is a port of the Unix game 'hack', by the Software Distillery, version 1.0.30.
FaOj	Strips garbage off Xmodem transfered object files	Juggler	Eric Graham's stunning HAM animation of a robot juggler	Fred Fish Disk 53		Fred Fish Disk 53	This is a port of the Unix game 'Lam', by the Software Distillery, version 1.0.30.
Handler	AmigaDOS handler (device) example from C-A	VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols	Fred Fish Disk 54		Fred Fish Disk 54	This is an official IFF specification disk from Commodore, an update to disk 16.
Hp-10c	Mimics a HP-10C calculator, written in Modula-2	Fred Fish Disk 48	Alpha version of a hard disk file archiver	Can	V2.05 of Matt Dillon's cash shell (Modified for Manx C). by Matt Dillon.	Fred Fish Disk 55	Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D
IFFEncode	Saves the screen as an IFF file	Bru	Version 1.30 of a terminal emulator with phone directories	NewStartups	New G Startup modules: with 1.2 files and better quote handling. opens a stdio window, using user specs. by Commodore,	Bawk	Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D
HDump	Dumps info about an IFF file	Comm	Version 2.04 of Matt Dillon's Unix 'ash'-like CLI replacement, including Lattice and Manx C source	AStartup.asm	Change another program's screen colors. by Carolyn Scheppe	MWB	Example for closing a custom Workbench screen. S-E-D
Jsh	BDS C-like CLI shell	Csh	Disk benchmark program for Unix and Amiga Computes disk storage of a file or directory	TWStartup.asm	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon	CosWB	Generates one-line fortune-cookie aphorisms. S-E-D
NewStat	STATUS-like program, shows priority, processes	Diskperf	Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery.	Palette	Save a normal or HAM mode screen as an IFF file. by Carolyn Scheppe	Cookie	Build-your-own mouse port clock. Creates C source files for menus, based on text descriptions. S-E-D
Revers	Game of Reversi, version 6.1	MemWatch	A real-time execution profiler for Manx C programs. Includes C source.	ShanghaiDemo	Demo of the Activation game Shanghai. A double-buffered sound example for Manx C. by Jim Goodnow	JTime	Build-your-own mouse port clock. Creates C source files for menus, based on text descriptions. S-E-D
Ucodec	Translate binary files to text, Unix- like programs	Profler		SoundExample	Working vspire example. by Eric Cotton	MenuBuilder	CBM tutorial on new packets and structures in AmigaDOS 1.2.
Vdraw	Drawing program, version 1.14			Vspires	V2.6 of Dave's V100 terminal emulator with kermit and xmodem. by Dave Wecker	PascalToC	Pascal to C translator, not so great. S-E-D
VoiceFiler	DX MIDI synthesizer voice filer program			V100		Prep	'hacker'-like FORTRAN preprocessor. S-E-D
Window	Example of creating a DOS window on a custom screen			Fred Fish Disk 56		RunBack	Starts programs from CLI allowing CLI window to close. E-D
Fred Fish Disk 39		Fred Fish Disk 49	Update of electronic spirograph from disk 27	CopBoard	Copboard device interface routines, to provide a standard interface. by Andy Finkle	SunMouse	This program automatically clicks in windows when the mouse is moved over them. Version 1.0, E-D
AnsiEcho	'echo', 'touch', 'list', 'ls' written in assembler.	Cydoos	Enhanced version of DrUtil from disk 35	ConPackets	Demos the use of DOS Packets, ConUnit, etc. by Carolyn Scheppe	Fred Fish Disk 56	Preliminary plans for a SCSI disk controller board.
Display	Displays HAM images from a ray- tracing program, with example pictures.	DrUtil	Scans a set of object modules and libraries searching for multiply defined symbols	GetDisks	Program to find all available disk device names and return them as an exec list. by Philip Lindsay	Asm58k	Macro assembler, version 1.0.1. E-D
Driver	Example device driver source, acts like RAM disk	MultDef	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process	GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck Molman	Assigned	Example for avoiding DOS internal disk requester, by scanning the list of assigned names. S-E-D
Xisp	XLisp 1.7, executable only	MyUpdate	Program to update system files with stripping comments from C header files, and interactive verification of the updating process	Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Scheppe	Dk	Pretends to set away at CLI window. S-E-D
Fred Fish Disk 40		Plot	Moire type pattern generator with color cycling Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.	MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Scheppe	Fip	Fips whole screen as a joke. S-E-D
Anost	Terminal emulator with Xmodem, Kermit, and C-B-B protocols, function keys, scripts, RLE graphics and conference mode.	Polygon	Moire type pattern generator with color cycling Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.	mCAD	An object oriented drawing program, V1.1 by Tim Mooney	Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.	OMouse	Example of setting the timestamp on a file, using a technique from Commodore-Amiga More extensive version of the bees program on disk 31	Fred Fish Disk 57		Free	Prints amount of free space on all drives. S-E-D
Arc	Popular file compression system, the standard for transferring files	Touch		Fred Fish Disk 58		MalloTest	malloc/free memory test program. S-E-D
AreaCode	Program that decodes area codes into state and locality.	Trees		ASDG-nd	Extremely useful shareware recoverable ram disk. by Perry Kivlowitz	Melt	Pretends to melt the screen. S-E-D
Bink	'link' replacement linker, version 6.5			BgView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson	Nart	Graphic flying string demo. S-E-D
Cosmo	Data General D-210 Terminal emulator			EGraph	Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurence Turner	Purdy	Easy way to set printer attributes from Workbench. E-D
Dp210	Windowed DOS interface program, V.1.4			HyperBase	Shareware data management system. V1.5 Walks through the free memory lists, zeroing free memory along the way. by John Hodgson	RayTracer	Simple ray tracing program. E-D
DrUtil	Windowed AmigaDOS CLI help program			MemClear	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgson	SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
DOSHelper	Prints text files with headers, page breaks, line numbers			NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgson	SnapShot	Memory resident screen dump. E-D
PagePrint	Prints text files with headers, page breaks, line numbers			RainBow	A Mauser-style rainbow generator. by John Hodgson	TagBBS	Shareware BBS system, version 1.02
PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Vessard, with Sprite Editor edits two sprites at a time			SMUSPlayers	Two SMUS files, to play SMUS IFF music formatted files. by John Hodgson	Fred Fish Disk 57	Shareware disk cataloging program.
SpriteEd	With a screen-saver feature. Vessard, with Sprite Editor edits two sprites at a time			View	A tiny ILM viewer by John Hodgson	AmiCat	Shareware intuition spelling checker, V2.0. E-D
X-Spell	Spelling checker allows edits to files			WBump	JX-80 optimized workbench printer that does not use DumpPort by John Hodgson	AmigaSpell	Shareware intuition spelling checker, V2.0. E-D
Fred Fish Disk 41				Fred Fish Disk 51		Bouncer	3-D bouncing ball written in MultiFort. S-E-D
AmigaVenture	Create your own text adventure programs in AmigaBasic			Compress	Update to the file compression program on disk 6	Comm	Terminal program version 1.33, E-D
Csh	Version 2.03 of Dillon's Csh-like shell. Executable only			Cos	'Wheel of Fortune'-type game in AmigaBasic	Dus5	Hex, octal, & decimal calculator. E-D
Dbug	Macro based C debugging package V#2 example from CBM, update to intuition manual			DrSaid	Unix-like 'ls' and 'lsd' for finding the differences between two files, and then recreating the other, given one file, and the list of differences.	HexCalc	Hex, octal, & decimal calculator. E-D
DuPlayField	Manual from CBM, update to intuition manual			So, Usq	Portable versions of the CPM squeeze and unsqueeze	Icons	Various big and alternate image icons.
GeFile	Heath's file requester, with source			Fred Fish Disk 52		Mandala	Mandala graphics and sound. E-D
LatXref	Cross reference of Lattice 3.10 header files			Assign	Replacement for AmigaDOS 'assign' command in C	PerMail	Demo shareware personal file manager.
Lines	Line drawing demo program			Fractal	Makes random fractal terrains	RLClock	Menu bar clock version 1.3. E-D
SoftFont	Changes font used in a CLI window			Poly, HAMPoly	Workbench-type demos for making polygons in lores and HAM	RTGates	Graphics demo of 3D cubes. E-D
V100	Version 2.3 of the VT-100 terminal program.			McGads	Example of mutual exclusion gadgets with GadgetText	Wheel	'Wheel of Fortune'-type game, in AmigaBasic
Fred Fish Disk 42				Tek4010	Tektronix 4010 terminal emulator	Fred Fish Disk 58	This is version M610 of the MicroGNUMacros. Source and executable are included, as well as source for other computers besides the Amiga.
Fred Fish Disk 43				VDraw	Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program	Asm58k	Macro assembler, v1.0.3, E-D
BesicBong	AmigaBasic program demos page flipping of a 3D cube			Fred Fish Disk 53		BitLab	Bitler exploring program, in C, S-E-D
Bom	Demo copy of B.E.S.T. Business Management System.			Animations	Demo animations with player program for Aegis Animator	Conman	Replacement console device handler adds editing and history to any application that uses CON;. v0.9, E-D
BolList	A list of Amiga Bulletin Board Systems			ARCre	Creates rename scripts for files with long names, so they can be easily 'arced' and un'arced.	Console	Replacement console routines, in C, S-E-D
Cc	C compiler rewrites for Manx and Lattice C			ARP	Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'find', and 'ls'	Dk	Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D
Copper	A hardware copper list disassembler			Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It will produce simple assembly language output, but needs a lot of work.	Frag	Displays memory fragmentation by listing the size of free memory blocks, in C, S-E-D
InstIFF	Converts Instruments demo sounds to IFF sampled sounds			Spreadsheet	Port of program to split Unix 'tar' archives Utilities to encode and decode binary files for ASCII transmission, expanding them by 35 percent	IconType	Change the type of an icon, in C, S-E-D
PopColours	Adjust RGB colors of any screen			TarSplit		Make	'make' in Manx C, S-E-D
SpriteCook	Simple clock is displayed on a sprite above all screens			Ukencode		MonProc	Monitors processes for packet activity, in C, S-E-D
ST Emulator	Non-serious Atan ST emulator					MouseClock	Turns mouse pointer into a digital clock in C, S-E-D
WBrin	Lets Workbench programs be run from the CLI					Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Wld	Two Unix shell style wild card matching routines					Spew	Generates National Enquirer-type headlines from rules file. In C, S-E-D
Fred Fish Disk 44						Spool	Three programs to demonstrate multitasking and spooling in a printer spooler. In C, v1.2, S-E-D
Icons	Miscellaneous icons					Wc	Counts words as Unix 'wc', but faster, in C, S-E-D
NewIFF	New IFF material from CBM for sampled voice and music files					Fred Fish Disk 70	This is a disk of shareware programs.
RayTracePics	The famous ray-tracing pictures, from FF#39, now converted to IFF HAM for 'much' faster viewing.					AmigaMonitor	Explores state of the system, v1.13
format	Displays normal and HAM ILM files					BlackBook	Standard file compressor and librarian, v2.3, a port of MS-DOS v5.0. E-D
ViewILBM						DoTi	Phone book program.
Fred Fish Disk 45						GravityWars	Intuition-driven file manipulator program, v2.0.
Cue	Anchor 'make', with more features					Joss	Game of planets, ships and black holes, v1.03.
Make	Miscellaneous pictures						Alternate user interface to CLI and WB, v2.1.
Pictures	Updates an older disk with newer files from another disk						
Update	Searches a disk for files of given name						
WhereIs							
Fred Fish Disk 46							
Asm	Shareware 68010 macro assembler, ROM Kernel Manual compatible						
CheckModem	'execute' file program detects presence of modem						
Egad	Geoged editor from the Programmers Network						
Jive	Transforms a file from English to Jive.						
MyLib	A binary only copy of Ma's alternate						

Lens	Magnifies area around mouse, shows it in a window, v1.0.	HardCopy	Sends a transcript of a CLI session to a file, in C, S-E-D	Plot6	A star plotting program with source.	computes, and prints expressions. Includes transcendental function support. c Source included.	
Life-3d	3D version of the classic cellular-automaton game, v1.2.	MouseOff	Update to disk 73, turns off mouse pointer, S-E-D	RawIO	Example of setting raw mode on standard input.	by John Olan	
Logo	Logo language interpreter	SeiFont	Changes the font in a Workbench screen, v2.0, S-E-D	Rocket	Lunar Lander for Workbench, with source.	Two programs to pack and unpack shell archives (traditional Unset bundling of multiple text files for posting or electronic mail). Includes C source.	
SeiKey	Demo keypad editor, v1.0	SpeedDr	Another fast 'dir', in assembler, S-E-D	More	'more'-like text viewing utility, v1.0 with source.	by Fabian G. Dufre	
Vpg	Makes displays for aligning video monitors, v1.0	Fred Fish Disk 76 & 77	These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.	Views	Simple Unix news reader.	SmallLib	8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt.
Amiga Basic	Makes airfoils using the Joukowski transformation, in C, S-E-D	Fred Fish Disk 78	Cycles	AutoPointAuto-selects window under the mouse pointer, with screensaver.	UUencode	Encode/decode binary files for e-mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options.	
Books	Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program, file comparison utility, string search program, S-E-D	Cycles	Cycle game like 'Tron', v1.0, E-D	ClickToFront	Double-clicks in window brings it to front, v1.1, S-E-D	MicroEmacs	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Comm	A variation of 'lines', but with variable color blocks. E-D	EOMS	Experts Only Mercenary Simulator game, E-D	Cmd	V3.0 of a tool to redirect printer output to a file.	AutoIconOpen	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
DiskX	Great terminal program, v1.34, E-D	MandelVroom	Mandelbrot generator with enhanced palette controls, fixed floating point, presets, v1.50, in Manx, C, S-E-D	FileISG-Demo	Demo of Softwood File Ilog, a database manager with sound and graphics.	DropShadow	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Epic	Utility for exploring file system. E-D	Fred Fish Disk 79	AsmTools	Fred Fish Disk 87	Adventure system from Byte May 1987, v1.2 E-D	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
IconMk	Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D	AssignDev	CLI tools in assembler: echo, loadit, mounted, setace, why, S-E-D	AdvSys	Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Icons	Makes icons for files, v1.2a, E-D	AuxHandler	Give devices multiple names, in C, S-E-D	Claz	Converts IFF files to PostScript, V2.0, S-E-D	IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
NewFonts	Two new fonts: 'shalt18', an electronic circuit element font, and 'tbn5', a PC-like font.	Comm	Example of a dos handler that allows use of a CLI via the serial port. Includes source.	Comm	Converts IFF files to PostScript, V2.0, S-E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
PerCLI	An AmigaBASIC CLI shell program.	Cmd	Redirects printer output to a file, in C, S-E-D	Diff	Updates to disk 75 of Unix-like 'diff', S-E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
PWDemo	Demo of the commercial product PowerWindows, v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D	Info	AmigaDOS 'info' replacement, in C and assembler, S-E-D	Dme	V1.27 of Dillon's text editor, update to disk 74, E-D	Fred Fish Disk 95	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Rot	Creates and animates 3-D objects, v0.5, E-D	Kill	Removes a task and its resources, in C, S-E-D	DropShadow	V2.0 of program that puts shadows on Workbench, S-E-D	Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
TimeSet	Sets time from Workbench, E-D	MZError	Displays errors from TDI Module-2 complex, S-E-D	Eib	Shared library example in Manx C.	AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Fred Fish Disk 72	This is a disk of IFF pictures.	ManProc	Update to process packet program from disk 68, in C, S-E-D	ID-Handler	An AmigaDOS device handler that generates unique identifiers, V1.0, S-E-D	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Add	Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D	Mounted	Program for testing if a drive is present, in a script, in C, S-E-D	Install	Alternate AmigaDOS 'install' programs, S-E-D	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
AutoIconOpen	Fools WB into thinking mouse has double-clicked icons. In C, S-E-D	Nro	Another 'nro'-style text formatter, in C, S-E-D	MemWatch	Waits for low memory trashing, V2.0, S-E-D	IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Do	Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. in C, S-E-D	ParTask	Finds parent task, in C, S-E-D	MovePointer	Moves pointer to given location, S-E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Dsolve	Slowly displays IFF files, ala Nov 86 Dr. Dobbs' program. In C, S-E-D	QueryAny	For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D	MoveWindow	Moves window to given location, S-E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
DTerm	Flexible, reprogrammable terminal program v1.10, E-D	ScrSize	Resets preferences settings for screen size, in C, S-E-D	MunchingSq	Munching Squares hack, S-E-D	Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Expose	Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D	SharedLib	Example of a shared library, in C and assembler, S-E-D	PaText	Example shows test to see if this is a PAL machine, S-E-D	Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Lit	Scans a text file, converts to C-style printable strings, v2.0, S-E-D	Task	Simple CreateTask() example in C, S-E-D	Sc	Generates random scenery, S-E-D	AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Lmv	'Long Move', program views series of IFF pics in quick succession, up to 19 fps. Shareware, E-D	Uw	Unix Windows client v1.0, in C, S-E-D	TeX4855	TeX4855 printer driver	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
MouseOff	Mouse pointer disappears after ten seconds of non-use. In C, S-E-D	Who	Lists tasks on ready and wait queues, in C, S-E-D	WBduatPF	Example of dual-playfield screen, update to disk 41, S-E-D	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
ParOut	Examples of controlling parallel port with resources instead of the PAR device. In C, S-E-D	Fred Fish Disk 80	(see Fred Fish 80)	WarpText	Fast text rendering routines, S-E-D	IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
PerPalFont	Child-like font	Fred Fish Disk 81	(see Fred Fish 81)	Yaif	Example IFF reader, S-E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
RunBackGround	Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C, S-E-D	Asm68k	V1.1.0 of a macro assembler	Zoo	A file archiver like 'arc', v1.42A, E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
SnapShot	Screen dump utility, update FF 66 E-D	AutoFacc	Shrinks the FACC window and moves it to the back	Fred Fish Disk 88	(see Fred Fish 88)	Fred Fish Disk 95	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
TypeAndTel	Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D	Brushes	53 custom IFF brushes of electronic symbols	Fred Fish Disk 89	(see Fred Fish 89)	Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Xplor	Prints info about system lists, in assembler, S-E-D	CheckIFF	Checks structure of an IFF file. C and V1.4 update to disk 74 of a simple command line editor	Fred Fish Disk 90	(see Fred Fish 90)	AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Fred Fish Disk 74	Edits and recalls CLI commands, v1.3, E-D	Comman	Replaces console handler to add editing and history to many programs	FunKey	Shareware function key editor, V1.01, E-D	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Cred	Intercepts graphic printer dump calls and accesses color map, width, and screen resolution, C, S-E-D	Fonts	Miscellaneous fonts	MFF-Demo	Demo of MicroFiche File database program	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Dme	Simple WYSIWYG text editor for programmers, v1.25. Update of FF 59 E-D	Icon	V6.0 of the icon programming language	ScreenShift	Adjust screen position like Preferences, S-E-D	IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
DropShadow	Workbench dropshadows, v2.0. Update to disk 59, E-D	KeyLock	Freezes the keyboard and mouse until pass word entered.	Snake	Bouncing squiggly lines demo, S-E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Funds	AmigaBASIC program tracks mutual or stock p-D	ScatDisplay	hack created from 'img'	AutoEnguier	Screen contraption requester improvement, S-E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Less	Text viewing program, like Unix 'more', v1.1, update to disk 34, S-E-D	Smith	Smushes an IFF file.	Demolition	Display Hack S-E-D	Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Makemake	Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D	Smush	Smushes an IFF file.	Fred Fish Disk 92	(see Fred Fish 92)	Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
mCAD	Object-oriented drawing prog. v1.2.4, update to FF 59 Shareware, E-D	Target	Each mouse click becomes a gunshot	AmiGazer	Night sky viewer of 1573 stars, set date, time, day, E-D	AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Random	Simple random number generator in C, S-E-D	Fred Fish Disk 82	Adventure A part of the classic Crowther and Woods Adventure game	CrdFile	AmigaBASIC card file study aid, E-D	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
TDebug	Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D	AmicTerm	V0.50 of a telecommunications program, with scripts, media, beeps, enhanced file requester	Conman	Console handler replacement gives line editing and history to most progs, v0.98 E-D	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Units	Converts measurements in different units, includes 'chart' option, in C, S-E-D	D2D-Demo	Demo version of Disk-2-Disk from Central Coast Software	MandelVroom	Sight update to disk 78 Mandelbrot program, E-D	IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
XCOPY	Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D	DX-Synth	Voice file program for Yamaha DX synthesizers, update to disk 38	NewDemos	Replacements for lines and boxes demos that take less CPU time, E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Fred Fish Disk 75	Play with Bezier curves points and granularity, S-E-D	DiskMan	V1.0 of another D2D utility program	Othello	Game of Othello, E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
BSplines	Play with b-splines, as above, S-E-D	Icons	Miscellaneous new icons	PrinText	Displays text files with gadgets, speech, IFF display, v1.2, E-D	Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Comm	C-source for Comm terminal program v1.34, S-E-D	Parl	Universal MIDI patch panel, v1.2	PrnDnGen	Automatic printer driver generator, v2.2b, E-D	Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Copy	Replacement 'copy' command v1.0,	Rocket	Another Workbench hack, plays Lunar Lander	RainBench	Cycles colors of Workbench backdrop or text, E-D	AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
preserves	date, in C, S-E-D	Sand	Game of sands following your pointer.	ShorCut	Makes single-key shortcuts for entering commonly typed CLI commands, as well as custom macros. E-D	ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Diff	Simple 'diff' in C, S-E-D	Fred Fish Disk 83	This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.	ShowPrint	Displays and prints IFF pictures of all sizes, and controls printer output styles, v2.0 E-D	HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
DuM2	Another D2D utility in Module-2, v1.5, S-E-D	Fred Fish Disk 84	AudioTools Programs from Rob Peck's July/August Amiga World article	Sizzlers	Graphics demos, v1.7.0, E-D	ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Eless	Fast 'dir' program in C, S-E-D	Ed	Simple editor, similar to Unix 'ed', based on the editor in Software Tools.	Timer	Small Workbench timer counts time and /s/ minute, E-D	Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
Fd	Faster 'less' in C, S-E-D	GravityWars	Game of planets, ships and black holes, v1.04, update to disk 70.	Tools	Horvath's tools: a memory editor, memory disassembler, ASCII chart, and calculator. E	Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		HunkPad	Adds legal padding to executables for Xmodem transmission.			Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		PipeHandler	An AmigaDOS pipe device which supports named pipes and taps. V1.2			AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		PopCLI	V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.			ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		Requester	Update to disk 34 of a file requester similar to DPaint.			HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		ScottDevice	V33.1 of a 'mountable' MicroForge SCSI driver.			IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		Viacom	Another Schwab hack, makes TV-like static on screen.			ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		Fred Fish Disk 85	V2.06 of Dillon's 'sch'-like shell			Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		Cah	Source to wildcard file requester			Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		HideReq	Hides expansion memory from programs			Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		Hide	Shareware tools to manipulate IFF images			AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		ImageTools	Shareware tools to manipulate IFF images			ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
		LowMem	Server/Shared library to aid in low memory situations			HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Terrain3d	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Fred Fish Disk 96	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						Cmd	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						AutoTools	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ClickUpFront	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						HeliosMouse	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						IFF2Ps	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tie-line statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon.
						ModuloTools	Version 1.27 WYSIWYG programmer editor. Not a word processor

INDEX OF ADVERTISERS

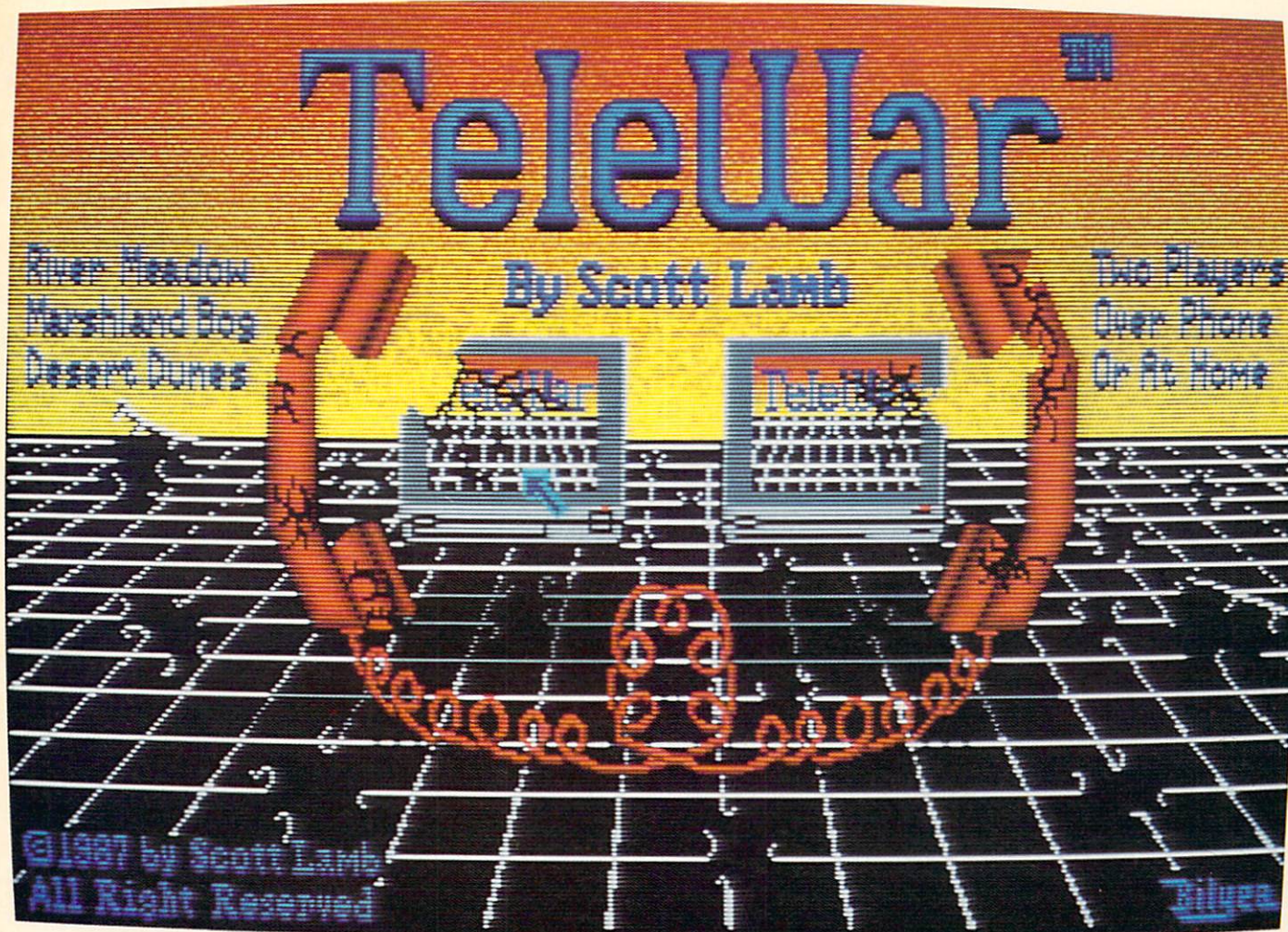
A-Squared Distributions, Inc.	17
Absoft	94
Aegis Development	56,57
Aminetics	70
Applied Visions	CII
Bethesda Software	49
Brown-Wagh Publishing	31,33,35,37,39
Byte by Byte	CIV
Comp-U-Save	100
Computer Systems Associates, Inc.	71
ComputerMart	78
Data Solutions	53
DataMax Research	27
Discovery Software International, Inc.	64,65
Firebird	63
Fuller Computer Systems, Inc.	75
Haitex Resources	23
Hilton Android	102
HyperTek/Silicon Springs	68
InterComputing, Inc.	44
Kent Engineering & Design	106
KlineTronics	108
Lattice, Inc.	7
Lightning Publishing	22
Manx C	15
Megatronics	9
Meridian Software	43
Microbotics	11
Microillusions	CIII
NewTek	1
Newwave Software	107
Pacific Peripherals	19
Palomar Peripherals	95
Peacock Systems	12
Pim Publications	103-105,128
Pioneer Computing	116
Prolific, Inc.	3
Right Answers Group	81
Second Source Systems	101
Sedona Software	6
Snake Design Software	58
Soft-Link	83
Software Integration Solutions	115
Software Supermarket	92
Software Terminal	127
Software Visions	55
Speech Systems	50
Spencer Organization, Inc.	113
Sunsmile Software	16
The Memory Location	74
The Other Guys	21
TRU-IMAGE	3
Westcom Industries	109

Fred Fish Disk #7	Replaces FF57 for Copywrite problems
OutAndPaste	Implementations of Unix cut and paste commands, by John Weald
GraphIt	Program to plot simple functions in 2 or 3 dimensions, by Flynn Fishman
Juggler	V1.2 of robot/juggler animation. Uses HAM mode and ray tracing, by Eric Graham
MouseReader	Shareware program to read text files and view FF files using only the mouse, by William Betz
Spines	Program to demonstrate curve fitting and rendering techniques, by Helene (Lee) Taran
Shm	Simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Ediss
Fred Fish Disk #8	
Access	16 color terminal program based on Comm V1.34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young, comm by D.J. James. E.
Backup	Writes AmigaDos disks as the backup destination. recovers files from the backup disk. Requires manual decisions on disk structure, by Alan Kent SE
DCDemo	DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time, by Ed Alford, MicroAse Software
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk, by Alan Kent SED
QBase	Quick-Base, a "Mail-Base Management Utility", define and maintain a maximum of 200 records per file, by Kevin Harise E
Thai	Thai language quiz program. Speak or type english/Thai sentences from supplied file, by Alan Kent SE
Fred Fish Disk #9	
A-Render Version 3	A Ray-Tracing Construction Set for the Amiga Computer, by Brian Reed ED
Fred Fish Disk 100	
Berserk	Must see animation, from Sept 87 FAUG meeting, by Leo Schwab
Conman	Console handler replacement, provides line editing and command line histories transparent to any application program that uses CON: windows. Shareware V1.0 by William Hawes. E.
WBlander	Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E
Fred Fish Disk 101	
CrPlane	Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertices. V1.0 by Thad Florian SE
IconAssembler	Change Workbench icons with IFF brush files by Stefan Lindahl E
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED
Midi	Midi library and utility set. Includes Midi monitor, routing utility, status utility, and more, by Bill Barton SED
Paintp	Postscript interpreter reads and previews postscript files on the screen, by Greg Lee S(assy)E
StartUps	Three C startup file replacements for standard Astartup.obj and LStartup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBSStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLISStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable, by Bryce Nesbitt SE
Fred Fish Disk 102	
Doug	Machine independent macro based C debugging package. Update of FF41, by Fred Fish profiling support by Binayak Banerjee SE
Match-stuff	Heavy duty text pattern matching stuff, includes simple match text replacement capability, by Pete Goodeve
Sectorama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume, by David Joiner E
SilCon	Smart input line interpreter with window for full editing. Upgrade of FF50 by Pete Goodeve. E
Xcon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31, by Pete Goodeve E
Fred Fish Disk 103	
ArtTrees	Library and test prog. implement routines for creating and using trees held in memory S.
Calc	A programmable RPN calculator.
Cref	A C cross ref. prog. S.
DosKwik	A pair of progs. which allows you to save files to one or more floppies for quick loading. Doesn't store Dos format.
IntuDos	A prog. to improve control and handling of the material on all disks in "CLI-area".

MFF-Update	A text import util. for MicroFiche. Files/demos on FF 83, and updates to some PD disk library databases.
Pack-It	Takes all files the files and dirs. on a disk and packs them into a single file, for sending via modem.
Sol	Amiga version of Solitaire.
Fred Fish Disk 104	
AnalysCalc	Is a large and powerful spreadsheet prog.
Fred Fish Disk 105	
AsmProgs	Some misc. assembly tools. Includes some S.
BasicProgs	LeastSquare solves least square probes and graphs results. S.
Bison	A replacement for unix " yacc" command. S.
Dmouse	Another prog in the tradition of display hacks". S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.
GravityWars	Game of planets, ships and black holes, v2.0 update to disk 84.
IPo2C	A util. to write a C-lang definition to mimic the instruction pointer S.
Pere-et-Fil	Ex. of creating and using reentrant processes. S.
Record Replay	Similar to "Journal" v2.0 update to disk 95.
Fred Fish Disk 106	
FuncKey	Shareware function key editor, v1.1 update to disk 89. Source avail. from author (Anson Man).
MoreArt	A small selection of some Amiga artwork.
QuickFlx	An IFF slideshow and cel animation program, v0.13.
RistNolla	A Finnish game. Also called Go-Moku. v1.0
Fred Fish Disk 107	
Csh	V2.07 of Matt Dillon's csh like shell. S.
Dff	A util., similar to other common "diff" programs. S.
ProSuite	Suite provides ex. code of facilities such as FileIO, Requester, XText, DoRequest and a tutorial on how to program the Amiga. Book 1.01. S.
SVTools	Some useful tools. S.
Fred Fish Disk 108	
Alit	Dr listing prog. based on LD4 pg S
DrMaster	Disk cataloger. v1.0b, update to disk 89. S.
Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit. installed. S.
MoniDCMP	Lets you monitor the IntuiMessages that pass through an DCMP window. Prints the message class, mouse coordinates, qualifier values. Great for debugging. S.
PrintPop	A util. to send common control settings to the PRT: device. S.
Sectrama	Utilities to recover lost or damaged data from floppies and hard disks. v1.1, an update to disk 102.
Tek	V100 emulator for a Tektronix 4010/4014. (V2.6) update to disk 52. S.
Zoo	A file archiver, much like "arc". v1.24B, update to disk 87
Fred Fish Disk 109	
Machine	A new animation.
Sim CPM	A CPM sim. simulates 8080 along with i19 emulator. S.
UUpc	Allows you to hook up your Amiga as a usernet node. S.
Fred Fish Disk 110	
A68k	A 68000 assembler written in C. S.
Poc	An optimizing C compiler for the 68000 processor. update to disk 53, but not based on the code of that disk.
To Be Continued.....	
In Conclusion	To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.

•AC•

**To Order
Public Domain
Software,
please use the form on
page 128.**



THE ULTIMATE STRATEGY WARGAME

TeleWar allows you to use your computer and modem to play the ultimate strategy wargame over the phone, or on one computer in your home for \$39.95.



RIVER MEADOW

MARSHLAND BOG

DESERT DUNE

- Superb graphic war simulations
- 12 war scenarios
- Animated artillery, armor, and supply units with air support
- Smooth scrolling graphics
- 3D perspective terrain maps
- All moves graphically enacted on tele-connected computer
- Digitized sound effects
- Easy to use menus and requesters

Make Telecomputing A Blast You'll Love TeleWar!

Published by Software Terminal
3014 Alta Mere, Fort Worth, TX 76116
817-244-4150 Modem: 817-244-4151

Available At Your Local Dealer

- Compatible with any modem
- 300, 1200, 2400, 9600 band
- Call originate or answer
- Null modem connect option
- Save game and transmit game options
- Opponent File Directories
- Send and receive typed messages with your human opponent
- Fully copyable to hard disk
- Upgrades available on BBS

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name _____
 Street _____
 City _____ St. _____ Zip _____
 Amount Enclosed _____

Please circle the appropriate item: New Subscription Renewal

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription. I have enclosed \$24.00 for 12 issues in the U.S. (\$30.00 Canada and Mexico, \$35.00 overseas). All funds must be in U.S. Currency on a U.S. Bank

Back Issues: \$4.00 each
 (foreign orders add \$1.00 each for Postage and Handling)

Please circle your Back issue choices below:

Vol1.1	Vol1.2	Vol1.3	Vol1.4	Vol1.5	Vol1.6	Vol1.7	Vol1.8	Vol1.9	Vol2.1	Vol2.2
Vol2.3	Vol2.4	Vol2.5	Vol2.6	Vol2.7	Vol2.8	Vol2.9	Vol2.10	Vol2.11	Vol2.12	

Public Domain Software:

\$6.00 each for subscribers (yes, even the new ones!)
 \$7.00 each for non subscribers

Please circle your Public Domain Software choices below:

Amicus:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21
A22	A23	A24	A25	A26						

Fred Fish:

FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11
FF12	FF13	FF14	FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22
FF23	FF24	FF25	FF26	FF27	FF28	FF29	FF30	FF31	FF32	FF33
FF34	FF35	FF36	FF37	FF38	FF39	FF40	FF41	FF42	FF43	FF44
FF45	FF46	FF47	FF48	FF49	FF50	FF51	FF52	FF53	FF54	FF55
FF56	FF57	FF58	FF59	FF60	FF61	FF62	FF63	FF64	FF65	FF66
FF67	FF68	FF69	FF70	FF71	FF72	FF73	FF74	FF75	FF76	FF77
FF78	FF79	FF80	FF81	FF82	FF83	FF84	FF85	FF86	FF87	FF88
FF89	FF90	FF91	FF92	FF93	FF94	FF95	FF96	FF97	FF98	FF99
FF100	FF101	FF102	FF103	FF104	FF105	FF106	FF107	FF108	FF109	FF110

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
 P.O.Box 869
 Fall River, MA 02722

Please allow 4 to 6 weeks for delivery

EBONSTAR

EbonStar is swift action competition wrapped in an electrifying arcade style format. Exceptionally unique and challenging, EbonStar will thrill you with it's dramatic sound, realistic graphics, and options to four players and man vs man or machine. Seize the challenge of EbonStar, pick up your copy today!



The warp's fathomless depths



The warp begins to explode as opponent is shot down



ROMANTIC ENCOUNTERS

A true to life adult encounter for men or women! Permit your wildest imagination's to interact with the enticing scenarios of the DOME. At the DOME, you can set yourself up for what ever you are after... We promise you'll get what's coming to you!



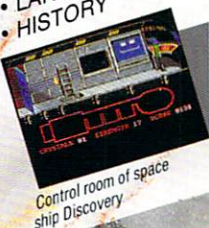
One of the Dome's alluring stories



The lounge, a place of enticing rendezvous

DISCOVERY EXPANSION DISK

Learning is an exciting adventure with DISCOVERY! Now you can expand your adventure with Discovery Expansion Disks: Math 1 • MATH CONCEPTS • SPELLING 1 • SPELLING 2 • TRIVIA 1 • TRIVIA 2 • SCIENCE • GEOGRAPHY • LANGUAGE • SOCIAL STUDIES • HISTORY



Control room of space ship Discovery



Working a solution to a math problem.

OTHER PRODUCTS FROM MICROILLUSIONS

- **PLANETARIUM™**
For the Serious Student of Astronomy
- **LAND OF LEGENDS™**
The ultimate dungeon adventure!
- **ONE TO ONE SERIES™**
Fire Power™, Galactic Invasion and Turbo™
- **BLACKJACK ACADEMY™**
Everything you ever wanted to know about Blackjack
- **THE FAERY TALE ADVENTURE™**
Amiga's number one adventure game!

All Microillusions Entertainment and Education products are developed especially for and available on Amiga, C64/128, MAC, Apple IIGS, and PC/MS DOS Formats.



17408 Chatsworth St., Granada Hills, Ca. 91344
Inside CA 818/360-3715 • Outside CA 800/522-2041 • FAX 818/360-1464

ROME WASN'T BUILT IN A DAY, UNTIL NOW...

Create your own universe with **SCULPT 3-D™**

SCULPT 3-D brings the power of 3 dimensional solid modeling and ray tracing to the Amiga. Imagine an image: choose a color, a shape, a texture. Spin it, rotate it, extrude it into the third dimension. Pick a camera lens, set your lights, and let SCULPT 3-D create a three dimensional picture complete with shadows, reflections, and smooth shading. All in 4096 colors with true edge to edge overscan video. Easily! Automatically! Change your mind? Change the colors, textures, camera or lights in seconds and create a new image. The only limits are the boundaries of your imagination.

"I haven't had this much fun with a program since Deluxe Paint II." John Foust of Amazing Computing.

"Performance previously only available on mini and mainframe computers." Info Magazine.



Now animate your universe with **ANIMATE 3-D™**

Enter the fourth dimension, time. Choreograph the free flowing and simultaneous movement of objects, lights and camera through space and time. Details of object rotation, camera movements, timing and action are controlled in an easy to use graphical interface or through a simple script language. Individual objects can be linked to orchestrate complex hierarchical movements that simulate live action. Quick check wireframe playback preview's your final production: storable as a compressed animation file playable from RAM, or recorded on videotape. Additional output options include single frame VCR control or image rendering to a frame buffer card. Animations can incorporate either solid modeling or ray tracing. ANIMATE 3-D is quite simply the most powerful and easy to use animation program available for the Amiga.

Expand your universe with the **BYTE BOX™**

Your Amiga 500 deserves the best you can give it. More memory for more powerful applications, faster performance, better graphics, and RAM disk storage. It deserves a memory expansion system that lets you add additional memory as you need it. In easy to install and easy to afford increments. The included memory verify software provides a visual check whenever you add additional RAM. The BYTE BOX is available in a variety of configurations from 0MBytes to 2MBytes of RAM.

- Easy to install
- Fully Auto-Configure
- Fast memory that's truly fast
- Has its own power supply
- Fully tested and ready to use
- Zero wait state design
- Low profile case
- Memory check software




BYTE by BYTE
CORPORATION

Aboretum Plaza II 9442 Capital of Texas Highway North Suite 150 Austin, TX 78759 (512) 343-4357

SCULPT 3-D, ANIMATE 3-D, and BYTE BOX are trademarks of Byte by Byte Corporation.
Amiga is a trademark of Commodore-Amiga, Inc. Deluxe Paint II is a trademark of Electronic Arts.